Efficient Strategy Learning by Decoupling Searching and Pathfinding for Object Navigation

Yanwei Zheng, Shaopu Feng, Bowen Huang, Chuanlin Lan, Xiao Zhang, Dongxiao Yu*

***Abstract*— Inspired by human-like behaviors for navigation: first searching to explore unknown areas before discovering the target, and then the pathfinding of moving towards the discovered target, recent studies design parallel submodules to achieve different functions in the searching and pathfinding stages, while ignoring the differences in reward signals between the two stages. As a result, these models often cannot be fully trained or are overfitting on training scenes. Another bottleneck that restricts agents from learning two-stage strategies is spatial perception ability, since the studies used generic visual encoders without considering the depth information of navigation scenes. To release the potential of the model on strategy learning, we propose the Two-Stage Reward Mechanism (TSRM) for object navigation that decouples the searching and pathfinding behaviours in an episode, enabling the agent to explore larger area in searching stage and seek the optimal path in pathfinding stage. Also, we propose a pretraining method Depth Enhanced Masked Autoencoders (DE-MAE) that enables agent to determine explored and unexplored areas during the searching stage, locate target object and plan paths during the pathfinding stage more accurately. In addition, we propose a new metric of Searching Success weighted by Searching Path Length (SSSPL) that assesses agent's searching ability and exploring efficiency. Finally, we evaluated our method on AI2-Thor and RoboTHOR extensively and demonstrated it can outperform the state-of-the-art (SOTA) methods in both the success rate and the navigation efficiency.**

I. Introduction

Object navigation[1], [2], [3], [4] requires an agent to seek and move to the vicinity of the target object in an unfamilar environment, leveraging panoramic or egocentric views. Benefiting from the development of artificial intelligence, end-to-end methods[3], [5], [6] based on reinforcement learning[7], [8] are widely used in navigation tasks owing to their flexibility and adaptability, with the expectation to approach and even surpass human expert navigation performance. Since humans often exhibit different motivations and strategies before and after discovering the target, which inspires the studies of navigation agent.

Prior studies have focused on developing efficient strategies for various stages by enhancing network structures. These enhancements have included the creation of modules tailored for specific purposes, such as searching based on object correlation [9], [4], [10], [11] and pathfinding utilizing target object features[12], [13]. While these improvements in model structure have led to enhanced performance, the increasing complexity of network architectures has posed new challenges. Specifically, the traditional sparse rewards used in training have exacerbated the difficulty of model convergence[1]. Furthermore, relying solely on rewards based on the shortest path can result in overfitting to the training scenarios. In essence, the adoption of single-stage reward mechanisms overlooks the causal relationships between strategies employed at different stages, thereby limiting the model's ability to adapt to diverse situations. Another critical bottleneck in enabling agents to learn effective strategies lies in inadequate spatial perception, particularly with regard to depth information. This limitation stems from the use of generic visual encoders, rather than using encoders specifically designed for navigation tasks, as the front-end of the model. Without this capability, agents are unable to discern which areas they have explored or unexplored, assess the distance to discovered targets, or plan viable paths to these targets. Consequently, a significant challenge remains in equipping agents with the ability to extract spatial depth information from monocular RGB images. Lastly, despite various metrics have been proposed, there is still short of practical indicators for evaluating the exploration efficiency of the agents during the searching stage.

To release the potential of the models in strategy learning, this paper introduces a Two-Stage Reward Mechanism (TSRM) tailored for object navigation. During training, each episode is divided into the initial searching stage and the

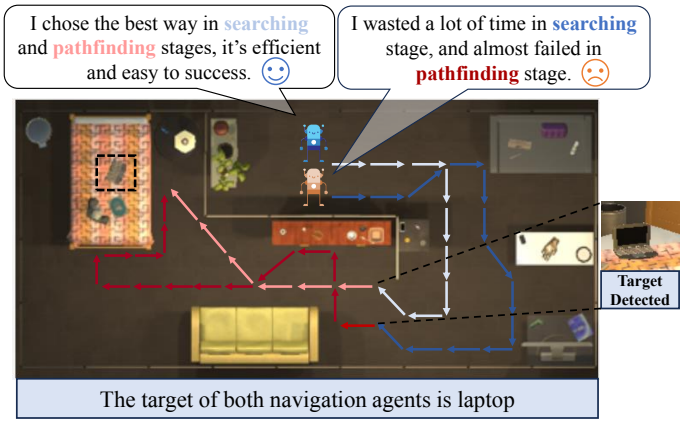

Fig. 1. The core idea of our method. Existing methods have proposed the concept of two-stage navigation for agents, but the sparse and improper reward signals make it difficult for agents to select an efficient and successful one from countless paths and actions. Our method utilizes a two-stage reinforcement learning reward mechanism and perception enhancement of spatial depth information to guide agents to learn efficient exploration and navigation strategies, improving success rates.

Yanwei Zheng, Shaopu Feng, Bowen Huang,Chuanlin Lan, Xiao Zhang, Dongxiao Yu are with Shandong University, Qingdao 266237, China.

E-mail: {zhengyw@sdu.edu.cn, fengsp@mail.sdu.edu.cn, huangbw@mail.sdu.edu.cn, lancl@sdu.edu.cn, xiaozhang@sdu.edu.cn, dxyu@sdu.edu.cn}. Dongxiao Yu is corresponding author (*).

subsequent pathfinding stage. The boundary between the stages is determined by the first observation frame of the target, achieved with a predefined level of confidence. In searching stage, the reward is proportional to the area of the newly observed region, encouraging the agent to explore more effectively. In the pathfinding stage, the mechanism not only rewards or penalizes movement actions based on the agent's distance to the target but also takes into account the impact of rotation actions on reaching the target. This consideration is particularly crucial for agents with limited perspectives, which have often been overlooked in previous approaches. Furthermore, to equip agents with a sharp "eye" for exploration and pathfinding, we propose a novel pretraining method tailored for navigation image encoders: Depth Enhanced Masked AutoEncoder (DE-MAE). . In this method, the RGB channels of the images serve as the input with certain patches randomly masked. Different from the classic Masked AutoEncoder (MAE)[14], the decoder is designed to reconstruct both the RGB and depth information. Compared with previous pretraining algorithms, the encoder pretrained by DE-MAE excels at capturing the spatial information within images, meeting the requirements of indoor scene navigation tasks. Lastly, we introduce a new evaluation metric, Searching Success weighted by Searching Path Length (SSSPL) that assesses agent's searching and exploring efficiency by comparing its performance to the theoretically optimal path of searching stage, compensating for the lack of comparability in previous indicators[15], [16]. The overall structure of our model is shown in Figure 2. And our contributions can be summarized as follows:

1) We proposed the Two-Stage Reward Mechanism (TSRM), which encourages navigation agents to efficiently explore the unknown areas before discovering the target, and then to accurately locate the target and find feasible path to reach it.

2) We propose Depth Enhanced Masked AutoEncoder (DE-MAE), a self-supervised algorithms to train the image feature extractor for the navigation agent.

3) A new evaluation metric named Searching Success weighted by Searching Path Length (SSSPL) is proposed to evaluate the exploring efficiency in searching stage.

II. Related Works

End-to-end navigation. End-to-end methods based on reinforcement learning have gained prominence in navigation tasks due to their flexibility across diverse environment. Initial investigations employed convolutional neural network to encode image observation, coupling them with recurrent neural networks to serve as the agent's memory mechanism[1]. As research progressed, researchers have realized the instructive role of the correlation between the positions of different objects in navigation decision-making, such as using graph neural networks to extract the distribution between different types of target objects[9], [17], [10], and constructing hierarchical relationships from objects, regions to scenes[5]. Furthermore, transformer [18] networks have seen widespread application in visual navigation tasks. They have been utilized both as feature extractors for images [2], [3] and as replacements for RNNs, capitalizing on their strengths in long-term memory and reasoning capabilities [19], [20], [21]. Drawing inspiration from human navigation strategies, recent studies [12], [13] introduced the navigation thinking network. This innovative approach selectively retains high-confidence features of target objects, derived from DETR [22], to aid the agent in pinpointing the target object's location.

Self-supervised learning in visual navigation. Self-supervised learning has demonstrated its capacity to extract valuable knowledge from extensive unlabeled datasets[23], [14], [24], [25]. In the domain of visual navigation, where images captured from the environment often lack annotations such as classification labels or textual descriptions, the application of self-supervised learning techniques for pretraining the image embedding extractor of navigation agents has gained considerable attention in recent years. For instance, contrastive learning approaches has been employed to enhance navigation performance under conditions of sparse rewards [26], [27], Similarly, OVRL-V2 [28] utilizes the Masked Autoencoder (MAE) for pretraining purposes. Furthermore, VC-1 [29] also leverages MAE and has built a substantial dataset tailored for universal embodied intelligence tasks, encompassing navigation, to train the image encoder.

III. Methods

A. Problem Setting

The agent is initialized to a random reachable position in the scene with random pitch and yaw angles: $s = (x, y, \theta, \beta)$. A random target g is assigned. According to the RGB image o_t and target g, the agent learns a navigation strategy $\pi(a_t|o_t, g)$, where $a_t \in A = \{MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, Done\}$. The output of $Done$ means the end of episode. Ultimately, When the agent is within 1 meter of the target object and meanwhile target object appears in the agent's view, if $Done$ is output the episode is considered successful. If the agent outputs $Done$ without reaching the target or the total number of output actions exceeds the predefined maximum episode length, the navigation episode is failed.

B. Two Stage Reinforcement Learning

When humans performs an object navigation task in an unknown room, they usually move around to explore at first and during exploration check whether the target object appears in the current view. If it does not appear, they continue searching for unexplored area. To enhance search efficiency, repeatedly searching the same area should be avoided. Once the target has been clearly observed, there is no need to continue searching. Instead, the focus shifts to locating the orientation of the target object and moving towards it. The two-stage learning approach for object navigation we propose aims to achieve this by employing a segmented design of the reward function, enabling the agent to learn the behavioral pattern of initially searching and subsequently finding a path to the target.

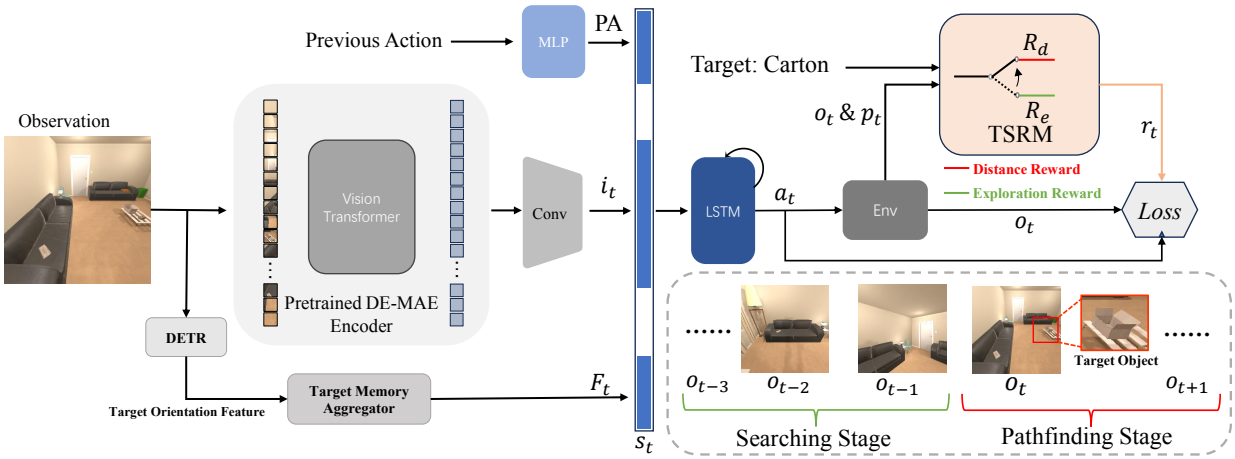

Fig. 2. Model overview. The agent obtains RGB image $\boldsymbol{o_t}$ and target orientation feature processed by DETR. Vision transformer pretrained by DE-MAE serves as the image feature extractor followed by a convolutional layer and outputs the image embedding feauture $\boldsymbol{i_t}$ based on $\boldsymbol{o_t}$. The target memory aggregator proposed by IOM[13] outputs target orientation embedding feature $\boldsymbol{F_t}$. $\boldsymbol{i_t}$, $\boldsymbol{F_t}$ and previous action embedding PA are concatenated as the state representation $\boldsymbol{s_t}$. Finally, the LSTM policy network outputs the action distribution $\boldsymbol{a_t}$ according to $\boldsymbol{s_t}$. The TSRM module determines the stage mode and the reward or punishment that should be given based on the historical and current location $\boldsymbol{p_t}$ and observation $\boldsymbol{o_t}$. When the target object is first detected, the episode irreversibly enters the pathfinding stage from the beginning searching stage, then distance reward will be given to agent replacing exploration reward.

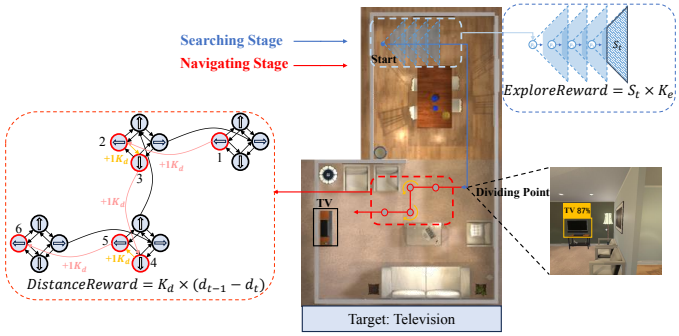

Fig. 3. Two-Stage Reward Mechanism. In the first stage, the exploration reward is proportional to the increase of the searched area which is the union of multiple trapezoids. After the target object is discovered, the navigation stage is entered. In this stage, each $Movehead$(pink) or $Rotation$(yellow) action will be rewarded or punished based on its impact to the distance between the agent and the target.

a) Searching Stage: An episode must begin with searching stage. In this stage, we set the exploration reward to be proportional to the increase of the overall clearly observed area. It is difficult to define the exact size of this area especially in different environments and observation equipment, so we propose an estimation approach. From the bird's-eye view, we simplify the area that can be observed clearly at each reachable position into an isosceles trapezoid intercepted from a right triangle (as shown in Figure 3).

When the agent reaches a new position, its searched area increases by an isosceles trapezoid. By performing geometric union operations on these trapezoids $T_{0,1...t}$, the resulting polygon corresponds to the area that the agent has searched. Since the polygon may include some areas out of the room, we perform a intersection operation on the polygon and the room's overhead projection. The final polygon is searched region RG_t. The size of the exploration reward is directly related to the area difference between RG_{t+1} generated after the agent takes a new action and RG_t in the previous step. RG_0 is initially null. Actually, we calculate searched region RG_t more efficiently recursively, and the specific formulas for calculating searched region and exploration reward are as follows:

$$RG_t = (RG_{t-1} \cup T_t) \cup RoomBounds \quad (1)$$

$$ExploreReward_t = K_e \times Area(RG_t - RG_{t-1}) \quad (2)$$

where K_e is a positve constant, $RoomBounds$ is the planar projection of the room, obtained from AI2-Thor's API, and $Area(\cdot)$ is a function that calculates the area of a plane figure.

b) Pathfinding Stage: If the target object can be clearly observed within current field of view, there is no need to search for unknown areas. Instead, the focus of navigation should shift to locating the target object and finding a feasible path to reach it. Previous navigation reward mechanisms have only considered changes in the Euclidean or geodesic distance from the agent's position to the target object, neglecting the impact of rotation actions that alter the agent's orientation. This is particularly crucial for agents without panoramic observation ability, as executing appropriate rotation actions at the right time is essential for subsequent movement. Conversely, making a wrong rotation action will not only cause the subsequent movement trajectory to deviate from the correct direction, but may also cause the target to disappear from the view.

As described in Fig 3, We process the scene as a grid map composed of passable position points, and convert the grid into a directed graph. Each directed edge in the graph corresponds to a $Moveahead$ or $Rotation$ action. The weights of all edges are 1. On this directed graph, the distance reward following the execution of an action is

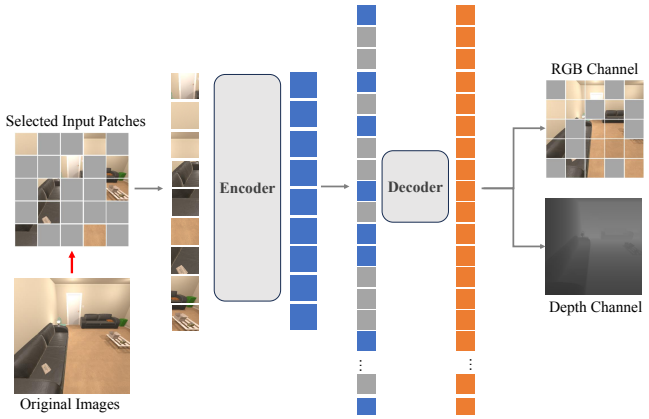

Fig. 4. **Depth Enhanced MAE (DE-MAE)**. During pretraining, a large random subset of RGB image patches is masked out. The encoder is applied to *visible patches*. The full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original RGB image and also predicts its depth image in pixels using two linear projection heads. After pretraining, the encoder is applied as image feature extractor for navigation tasks.

proportional to the reduction in the minimum distance to any potential successful terminal states . If this distance increases unexpectedly, a penalty is applied. The calculation formula for distance reward (penalty) is as follows:

$$DistanceReward_t = K_d \times (d(p_{t-1}, g) - d(p_t, g)) \quad (3)$$

where $d(p, g)$ is the minimum distance from position p to any terminal states that successfully reach the target g within 1 meter, and K_d is a positive constant. To improve computational efficiency, we use multi-source optimal path algorithm to preprocess the training scene grid data, thus avoid calculating the $d(p, g)$ during reinforcement training.

c) Dividing Point of Two Stages: Entering pathfinding stage too early can cause the model to overfit in the training scenes, underestimating the importance of exploration; conversely, an excessively long searching stage will reduce navigation efficiency and the probability of discovering targets.

On dividing point, the agent should objectively see the target object for the first time, and be aware of this subjectively. A threshold value C_{target} is set between 0 and 1. If in the detection results of DETR, the confidence of the corresponding target object in the current frame is greater than C_{target} for the first time, the agent enters the pathfinding stage from searching stage, the reward calculation mechanism also changes.

C. Depth Enhanced MAE

In previous studies, methods for pretraining image feature extractor include supervised learning like classification, or self-supervised learning such as Moco[23], MAE[14]. Although these pretraining algorithms are widely used, they ignore the particularity of navigation tasks and resulting in the loss of spatial information. Therefore, We propose depth enhanced MAE (DE-MAE), a self-supervised training algorithm improved upon standard MAE. As shown in Figure 4, this algorithm requires the decoder to reconstruct not only the masked RGB channels but also the depth channel of the original image.

a) Encoder and Decoder: We use the ViT-Base as encoder. The encoder receives an RGB image, and first converts it into patches, then add 2D sine-cosine positional embeddings after the linear projection. After multi-layer transformer operations, the feature map output by the encoder is used as the input of the decoder. The decoder employs a significantly smaller ViT network and takes both masked and non-masked patches as input. Finally, two linear projection heads are appended to map the decoder's output tokens to pixel-wise predictions for both RGB and depth channels of the corresponding image patches.

b) Loss: Our loss function computes the meansquared error (MSE) between the reconstructed and original images in the pixel space for both RGB and depth channel with identical weights on masked patches (RGB) and all patches (Depth).

c) Data Collection and Pretraining: We randomly sampled approximately 2 million RGBD images from training scenes of AI2-Thor. When collecting images, the camera intrinsics keep unchanged, and only horizontal flipping was employed for data augmentation. In subsequent experiments, we find that using additional augmentation techniques like random cropping will hinder the model's ability to learn the mapping relationship between image and real space, wasting the introduced depth information.

D. Navigation Based On RL

1) Policy Network: In the image feature extraction, the image feature $\boldsymbol{i_t}$ output by ViT pretrained using DE-MAE is obtained. To process the target orientation features obtained from DETR, we use the non-local target memory aggregation (NTWA) module proposed by IOM[13] to obtain the target orientation feature embedding $\boldsymbol{F_t}$. They are fused with previous action embedding PA, then the state representation $\boldsymbol{s_t}$ is obtained:

$$\boldsymbol{s_t} = cat\,(\,\boldsymbol{i_t}, \boldsymbol{F_t}, \boldsymbol{PA})\,\boldsymbol{W} \quad (4)$$

The LSTM[30] module is treated as a policy network $\pi\,(\boldsymbol{a_t}|\boldsymbol{s_t})$, and the asynchronous advantage actor-critic (A3C) algorithm[7] is used to train the model.

2) Reward Function: In addition to the two-stage reward mentioned above, the comprehensive supervision signal also incorporates rewards and penalties for collisions, navigation termination and other situations. The overall reward structure during reinforcement learning is outlined as follows:

(i) Exploration Reward: As described in Equation 2.

(ii) Distance Reward (penalty) : As described in Equation 3.

(iii) Collision penalty: When an agent collides for the first time, no penalty will be given. If the agent collides again in the same position and orientation, penalty of -0.1 will be given.

(iv) Slack penalty: In each step, a penalty of -0.01 will be given until the episode ends.

(v) Final Reward: A reward of +5 will be given when success.

IV. Experiments

A. Experimental Setting

1) Dataset: AI2-Thor and RoboTHOR datasets are selected for evaluation. AI2-Thor includes 4 types of room: kitchen, living room, bedroom, and bathroom, each consists of 30 floorplans, of which 20 rooms are used for training, 5 rooms for validation, 5 rooms for testing. RoboTHOR consists of 75 scenes, 60 of which are used for training and 15 for validation.

2) Evaluation Metrics: Success rate (SR), success weighted by path length (SPL)[15] metrics are used to evaluate our method. The formula of SR is $SR = \frac{1}{K}\sum_{i=1}^{K} Suc_i$, where K is the number of episodes, and Suc_i indicates whether the i-th episode is successful. SPL indicates the efficiency of the agent, its formula is $SPL = \frac{1}{K}\sum_{i=1}^{K} Suc_i \frac{L_i^*}{\max(L_i, L_i^*)}$, where L_i is the length of the path actually traveled by the agent. L_i^* is the optimal path length provided by the simulator.

In the ablation experiment, we uses three additional metrics to evaluate the performance at different stages, two of them are existing metrics Searching Success Rate (SSR)[16] and Navigation Success Weighted by Navigation Path Length (NSNPL)[16]. SSR is the success rate for the searching stage and is formulated as $SSR = \frac{1}{K}\sum_{i=1}^{K} Nav_i$, where Nav_i indicates whether the i-th episode enters the pathfinding stage. NSNPL measures the navigation capability of agents in the pathfinding stage, its formula is $NSNPL = \frac{1}{K_{Nav}}\sum_{i=1}^{K} Suc_i Nav_i \frac{L_i^{*Nav}}{max(L_i^{Nav}, L_i^{*Nav})}$, where K_{Nav} is the number of episodes that enters the pathfinding stage. L_i^{Nav} is the path length in the pathfinding stage and L_i^{*Nav} is the shortest path length in this stage which is calculated according to the starting position of the pathfinding stage. However, for the searching stage, SSR only considers the results, ignoring the efficiency of the searching and exploring process, and it's easy to achieve full scores for advanced methods[4], [16], [13], lacking comparability. Therefore we propose Searching Success Weighted by Searching Path Length (SSSPL) , a new metric to comprehensively evaluate both the search efficiency. Its formula is:

$$SSSPL = \frac{1}{K}\sum_{i=1}^{K} Nav_i \frac{L_i^{*Search}}{\max\left(L_i^{Search}, L_i^{*Search}\right)} \quad (5)$$

where L_i^{Search} is the actual path length in the searching stage and $L_i^{*Search}$ is the shortest path length in searching stage, it's calculated based on the initial position and all legal positions that satisfies the criterion of the two-stage dividing point.

3) Implementation Details: Our model is trained by 14 workers on 1 RTX 3090 Nvidia GPU. We first use DE-MAE in pretraining and then use reinforcement learning to train the agent for 1.5M episodes. By evaluating on the validation set, the target confidence threshold C_{target} for dividing point is set to 0.7. To calculate exploration reward, the distances from the agent center to the upper and lower bases of the observable trapezoid region are 1m (0.25m) and 4m (3m) respectively, in a straight-ahead (overhead) view. The learning rate of the optimizer is 10^{-4}. K_e and K_d are set to 0.1 and 0.15, and the dropout[31] rate of training is 0.45. During testing, the max episode length is set to 100. We show results for all targets (ALL) and a subset of targets ($L \geq 5$) whose optimal trajectory length is longer than 5.

TABLE I

Comparisons with the state-of-the-art methods on the AI2-Thor/RoboTHOR datasets

Method	ALL (%)		$L \geq 5$ (%)	
	SR	SPL	SR	SPL
SP[32]	64.70/56.82	40.13/29.71	54.80/53.72	37.35/30.95
SAVN[33]	65.68/58.14	40.60/31.47	54.92/54.03	38.41/30.56
SA[3]	68.37/60.49	41.04/31.58	56.33/55.74	38.96/31.08
ORG[9]	69.76/62.69	39.59/32.35	60.78/58.44	39.58/33.85
HOZ[5]	70.65/63.48	40.47/32.70	62.91/59.46	40.02/34.42
OMT[19]	72.37/66.49	33.20/33.76	61.00/65.85	30.72/36.55
G2SNet[34]	73.81/64.73	41.65/31.82	61.69/60.25	39.21/31.32
VTNet[2]	74.73/67.84	47.14/34.97	67.72/66.57	48.19/36.74
DOA[4]	80.81/71.75	46.85/36.58	74.25/69.30	48.69/39.15
DAT[12]	82.24/73.63	48.68/40.66	75.07/71.35	49.45/39.89
IOM[13]	83.42/73.95	49.38/39.15	76.10/71.28	49.94/39.52
MT[16]	85.47/73.86	47.02/38.46	80.08/71.85	50.26/38.74
Our	**86.73/74.93**	**50.48/40.37**	**81.71/73.36**	**52.09/39.10**

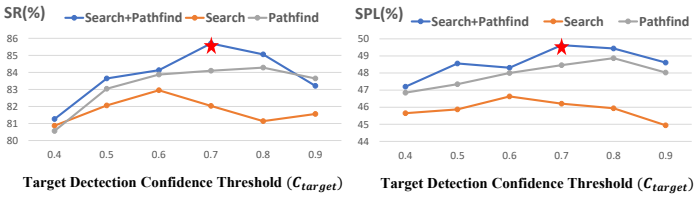

Fig. 5. We compared the navigation metrics using different target detection threshold values as two stages' dividing point on AI2-THOR validation set. The red star indicates the choices that optimize the given indicator.

B. Comparisons to the State-of-the-art

It can be seen from Table I that our method surpasses the previous SOTA method (MT) by 1.26/1.07, 3.46/1.91, 1.63/1.51, 1.83/0.36 (AI2-Thor/RoboTHOR,%) in SR and SPL. Our method is also superior to the other methods that attempt to implement two-stage strategies only based on network structure design, such as DAT[12] and IOM[13].

C. Ablation Experiments

To study the effectiveness of the Two-Stage Reward Mechanism and DE-MAE, we conducted ablation experiments on AI2-Thor as shown in Table II and Figure 5.

1) Baseline: Our baseline model uses the same Resnet18 feature extractor as previous methods[2], [13], [16], which is pretrained through the image classification task on ImageNet[35]. The two-stage reward mechanism is also not applied.

2) Two Stage Reinforcement Learning: As shown in Fig 5, we tried different target detection confidence threshold value. The excessively large threshold values result in long

TABLE II
ABLATION EXPERIMENT RESULTS ON AI2-THOR

Exploration Reward	Distance Reward	DE-MAE	ALL (%)					$L \geq 5$ (%)				
			SR	SPL	SSR	SSSPL	NSNPL	SR	SPL	SSR	SSSPL	NSNPL
			81.92	47.15	95.49	43.87	48.06	75.76	48.37	93.76	45.84	49.72
✓			82.95	46.63	96.63	46.06	48.23	77.74	48.18	95.21	47.39	49.68
	✓		84.28	48.87	95.98	45.63	50.40	78.93	50.74	94.83	46.95	52.38
✓	✓		85.70	49.63	97.06	47.08	51.56	80.46	51.61	96.81	47.92	53.45
		✓	83.21	49.35	96.60	46.20	50.29	77.97	51.40	95.12	47.28	52.63
✓	✓	✓	**86.73**	**50.48**	**97.79**	**47.56**	**52.56**	**81.71**	**52.09**	**97.03**	**49.20**	**54.10**

searching stages, causing the navigation inefficient and prone to missing target. While excessively low threshold may cause the agent to not fully explore the environment and navigate to the wrong target. Therefore, we ultimately set the threshold value C_{target} to 0.7.

In comparison to the baseline, the incorporation of the exploration reward has led to notable enhancements across multiple metrics, with particularly remarkable advancements in SSR (1.49/1.45, ALL/L$\geq$5, $\%$) and SSSPL (2.16/1.55, ALL/L$\geq$5, $\%$), indicating that the search accuracy and exploration efficiency have been significantly improved. Furthermore, upon integrating distance rewards into the baseline, the NSNPL metric, which serves as an indicator of pathfinding efficiency, has exhibited substantial gains (2.34/2.66, ALL/L$\geq$5, $\%$). When the exploration and distance rewards are combined, the metrics outperform those using either reward individually, demonstrating that rational search and pathfinding behaviors are complementary and cannot be substituted for one another.

3) Depth Enhanced MAE: When DE-MAE is independently integrated into the baseline, most metrics exhibit modest enhancements. This result suggests that, while the agent's perception capabilities is augmented, it doesn't know how to fully harness the ability for strategy learning. When we combine DE-MAE with TSRM that the improvement of perception serves as the foundation for strategy learning and the pursuit of maximizing exploration and distance rewards, leading to the best outcomes.

D. Irreplaceability of DE-MAE

TABLE III
COMPARISON EXPERIMENT RESULTS FOR DIFFERENT PRETRAINING METHOD ON AI2-THOR

Method(Backbone)	Pretraining Dataset	ALL (%)		$L \geq 5$ (%)	
		SR	SPL	SR	SPL
Baseline(ResNet18)	ImageNet	81.92	47.15	75.76	48.37
VC-1[29](ViT-Base)	Ego4D	82.12	48.19	76.46	49.76
Standard MAE(Vit-Base)	AI2-THOR	82.76	47.84	76.98	49.61
DE-MAE(ViT-Base) w/ crop	AI2-THOR	82.33	48.36	77.05	49.91
DE-MAE(ViT-Base) w/o crop	AI2-THOR	**83.21**	**49.35**	**77.97**	**51.40**

To demonstrate the irreplaceability of DE-MAE, we introduce other pretraining algorithms and data augmentation approach to the baseline model, the results are presented in Table III. In baseline, ResNet18 is pretrained by performing image classification task on ImageNet. The VC-1 uses the standard MAE method to pretrain ViT-Base, and its training dataset is Ego4D[36] (containing approximately 5.6 million egocentric RGB images). The DE-MAE applies on the same ViT-Base and uses about 2 million RGBD images from training scenes of AI2-Thor as training data. Standard MAE uses the same dataset as DE-MAE without depth channels. We have also tried DE-MAE with random cropping for data augmentation during pretraining.

The experimental results show that using VC-1 improves SR and SPL by 0.2% and 1.04% compared with baseline. This result indicates that the selection of the dataset does have an impact on the pretraining of the agent encoder, the results of Standard MAE approach also verify it. Though DE-MAE w/ crop uses the depth channel, its metrics has hardly improved compared to Standard MAE. While the DE-MAE not using cropping has increased the success rate by $0.45\%/0.99\%$ and the SPL by $1.51\%/1.79\%$ (ALL/L$\geq$5). Since we maintained the same camera intrinsic for navigation agents and pretraining image collection, random cropping may disrupt this consistency and hinder the model from learning the mapping relationship between images and real space, violating the purpose of introducing depth information.

V. CONCLUSION

In this article, we propose the Two-Stage Reward Mechanism (TSRM) for object navigation and the Depth Enhanced Masked Autoencoder (DE-MAE) to pretrain the image encoder, enabling the agent to learn effective exploration and pathfinding strategies through sufficient spatial perception capabilities. Moreever, to comprehensively evaluate the exploration ability and efficiency of agents, we have proposed a new metric Searching Success Weighted by Searching Path Length (SSSPL). The comparison experiments demonstrate that our work achieves state-of-the-art performance, the ablation experiments and analysis have also demonstrated the effectiveness and irreplaceability of our proposed TSRM and DE-MAE. Looking ahead, we believe that the sustained research on exploration and pathfinding strategies learning will catalyze further advancements in the field of object navigation.

VI. ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant

No. 2022YFF0712100, the National Natural Science Foundation of China under Grant 62202273, Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2022ZD02, Joint Key Funds of National Natural Science Foundation of China under Grant U23A20302, and the Key Technology Research and Industrialization Demonstration Projects of Qingdao under Grant 23-1-2-qljh-8-g

REFERENCES

[1] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3357–3364.

[2] H. Du, X. Yu, and L. Zheng, "Vtnet: Visual transformer network for object goal navigation," in *International Conference on Learning Representations*, 2020.

[3] B. Mayo, T. Hazan, and A. Tal, "Visual navigation with spatial attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 898–16 907.

[4] R. Dang, Z. Shi, L. Wang, Z. He, C. Liu, and Q. Chen, "Unbiased directed object attention graph for object navigation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3617–3627.

[5] S. Zhang, X. Song, Y. Bai, W. Li, Y. Chu, and S. Jiang, "Hierarchical object-to-zone graph for object navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 130–15 140.

[6] A. Pal, Y. Qiu, and H. Christensen, "Learning hierarchical relationships for object-goal navigation," in *Conference on Robot Learning*, 2021, pp. 517–528.

[7] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[9] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *European Conference on Computer Vision*, 2020.

[10] S. Y. Gadre, K. Ehsani, S. Song, and R. Mottaghi, "Continuous scene representations for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 849–14 859.

[11] M. Lingelbach, C. Li, M. Hwang, A. Kurenkov, A. Lou, R. Martín-Martín, R. Zhang, L. Fei-Fei, and J. Wu, "Task-driven graph attention for hierarchical relational object navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 886–893.

[12] R. Dang, L. Wang, Z. He, S. Su, C. Liu, and Q. Chen, "Search for or navigate to? dual adaptive thinking for object navigation," *arXiv preprint arXiv:2208.00553*, 2022.

[13] W. Xie, H. Jiang, S. Gu, and J. Xie, "Implicit obstacle map-driven indoor navigation model for robust obstacle avoidance," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 6785–6793.

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[15] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[16] R. Dang, L. Chen, L. Wang, Z. He, C. Liu, and Q. Chen, "Multiple thinking achieving meta-ability decoupling for object navigation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 6855–6872.

[17] W. Li, X. Song, Y. Bai, S. Zhang, and S. Jiang, "Ion: Instance-level object navigation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4343–4352.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] R. Fukushima, K. Ota, A. Kanezaki, Y. Sasaki, and Y. Yoshiyasu, "Object memory transformer for object goal navigation," in *International Conference on Robotics and Automation*, 2022, pp. 11 288–11 294.

[20] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 538–547.

[21] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in neural information processing systems*, vol. 34, pp. 5834–5847, 2021.

[22] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*, 2020, pp. 213–229.

[23] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[26] K. Yadav, R. Ramrakhya, A. Majumdar, V.-P. Berges, S. Kuhar, D. Batra, A. Baevski, and O. Maksymets, "Offline visual representation learning for embodied navigation," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[27] A. Majumdar, G. A. Sigurdsson, R. Piramuthu, J. Thomason, D. Batra, and G. S. Sukhatme, "Ssl enables learning from sparse rewards in image-goal navigation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 774–14 785.

[28] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav," *arXiv preprint arXiv:2303.07798*, 2023.

[29] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," *arXiv preprint arXiv:1810.06543*, 2018.

[33] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.

[34] W. Li, X. Wu, C. Wang, Z. He, P. Wang, and H. Wang, "Visual navigation by fusing object semantic feature," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2024, pp. 872–877.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[36] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.