TDANet: Target-Directed Attention Network For Object-Goal Visual Navigation With Zero-Shot Ability

Shiwei Lian[1], and Feitian Zhang[2]

***Abstract*—The generalization of the end-to-end deep reinforcement learning (DRL) for object-goal visual navigation is a long-standing challenge since object classes and placements vary in new test environments. Learning domain-independent visual representation is critical for enabling the trained DRL agent with the ability to generalize to unseen scenes and objects. In this letter, a target-directed attention network (TDANet) is proposed to learn the end-to-end object-goal visual navigation policy with zero-shot ability. TDANet features a novel target attention (TA) module that learns both the spatial and semantic relationships among objects to help TDANet focus on the most relevant observed objects to the target. With the Siamese architecture (SA) design, TDANet distinguishes the difference between the current and target states and generates the domain-independent visual representation. To evaluate the navigation performance of TDANet, extensive experiments are conducted in the AI2-THOR embodied AI environment. The simulation results demonstrate a strong generalization ability of TDANet to unseen scenes and target objects, with higher navigation success rate (SR) and success weighted by length (SPL) than other state-of-the-art models. TDANet is finally deployed on a wheeled robot in real scenes, demonstrating satisfactory generalization of TDANet to the real world.**

***Index Terms*—Vision-based navigation, reinforcement learning, autonomous agents.**

I. Introduction

THE objective of object-goal visual navigation is to find a target object in an environment using only egocentric visual observations. This task poses a major challenge to robots, requiring their visual understanding and inference of the complex scene to successfully locate a target instance.

Recent studies [1]–[9] have achieved great advancement in solving this problem using deep reinforcement learning (DRL) to train an end-to-end model. Learning an informative visual representation containing the relationships among objects is of crucial importance to the design of a robust navigation policy [1]. The learning of domain-independent feature encoding relationships among objects is a long-standing ill-posed problem [2] due to the existence of irrelevant information in RGB images such as background textures and colors. In addition, test environments vary in object classes and placements further complicating the deployment of the trained policy. Du *et al.* [3] and Fukushima *et al.* [4] applied visual transformer [10] to exploit the relationships of detected instances, which, however, at the same time increased the learning difficulty and computational cost. Other studies utilized prior knowledge graphs [5]–[7] or cosine similarity of word embeddings [4], [8] to assist in learning navigation strategies. These prior knowledge or representations of object relationships are rarely updated during training, thus limiting the adaptation to unseen scenes with dissimilar object placements. Moreover, most end-to-end models mainly focus on a limited class of target objects [9], while in the real household environment there may exist target objects not seen during training.

This letter investigates the generalization of the DRL-based object-goal visual navigation to both unseen test environments and unseen target objects by learning the relationships among objects. A target-directed attention network (TDANet) is proposed to train an end-to-end DRL policy for object-goal visual navigation with zero-shot ability. The network focuses on objects in the current visual observation that show strong correspondence with the target. A novel target attention (TA) module uses the information of the observed and target objects as well as their word embeddings as the input to learn both the spatial and semantic relationships between the target object and the detected objects in training. TDANet adopts the design of the Siamese architecture (SA) and distinguishes the difference between the current and desired states of the agent to guide the navigation, demonstrating strong zero-shot ability. The proposed model is evaluated in the AI2-THOR [11] embodied AI environment. The simulation results show that TDANet generalizes satisfactorily to unseen objects and scenes. In addition, comparison studies confirm that the proposed TDANet outperforms other state-of-the-art models with higher navigation success rate (SR) and success weighted by length (SPL). TDANet is also deployed in real scenes, showing its superior generalization to the real world.

II. Related Work

A. Object-Goal Visual Navigation

Object-goal visual navigation requires the agent to search for a target instance given only visual observations. Many end-to-end DRL models have been designed to establish the navigation strategy that maps the observations to actions for this task. Some studies learn object-goal visual navigation

This manuscript is accepted by IEEE Robotics and Automation Letters. *(Corresponding Author: Feitian Zhang.)*

[1]Shiwei Lian is with the Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University, Beijing, 100871, China lianshiwei@stu.pku.edu.cn

[2]Feitian Zhang is with the Department of Advanced Manufacturing and Robotics, and the State Key Laboratory of Turbulence and Complex Systems, College of Engineering, Peking University, Beijing, 100871, China feitian@pku.edu.cn

by learning implicit representations of the observation before inputting it into the navigation policy [12], [13]. Wortsman *et al.* [13] introduced self-adaptive visual navigation using meta-learning that learns to adapt to test environments without explicit supervision. Other studies exploit the object relationships or semantic contexts aiming for a more robust navigation policy [3]–[6], [8], [14]. For instance, Qiu *et al.* [6] and Li *et al.* [7] integrated hierarchical relationships among objects in the DRL model and achieved remarkable navigation performance using object detection outputs instead of raw RGB images. However, the prior knowledge graph used in these two methods is not updated during training, leading to limited generalizability of the model across different test scenes. Du *et al.* [3] introduced the powerful visual transformer to learn the relationship among objects without the use of a prior knowledge graph. Although improving the navigation performance in unseen test scenes, the visual transformer model significantly complicates computation and training and does not generalize to new classes of unseen objects.

In this letter, TDANet with a novel target attention module is proposed to learn both the semantic and spatial relationships between the target object and observed objects. TDANet is expected to be lightweight and generalizable to both unseen scenes and objects.

B. Zero-Shot Visual Navigation

Traditional object goal navigation only navigates to limited classes of target objects defined in the training set [15]. However, new classes of target objects will inevitably appear in the real household environment. Zero-shot navigation, which refers to navigating to objects not selected as the target object in training, has therefore attracted great research interest. While some recent studies, such as CoW [16] and VoroNav [17], realized zero-shot navigation with modular designs using large multimodal models, all those methods utilized depth images to generate a global map, which we consider absent from the observations of the agent of interest, thus out of the scope of this letter.

To improve the zero-shot ability of end-to-end DRL navigation models, some efforts have been made through the design of networks [12] or input encodings [9], [18]–[20]. For instance, Zhu *et al.* [12] proposed a Siamese network [21] for image-goal navigation across different scenes and target images. Khandelwal *et al.* [19] utilized the zero-shot ability of CLIP [22] to generate semantic embeddings of the goal for navigation. Although the use of CLIP improved zero-shot navigation performance, such a model design has fewer trainable parameters and thus impairs the learning of seen objects in the training set. Xu et al. [20] proposed the Aligning Knowledge Graph with Visual Perception (AKGVP) method leveraging the image-text matching ability of CLIP, achieving remarkable zero-shot navigation capability. Zhao *et al.* [9] proposed SSNet that used cosine semantic similarities and object detection results as the input to the navigation policy to eliminate class-related features. However, the detection matrix used in SSNet is still limited to predefined object classes, impeding further generalization of zero-shot navigation to unseen objects.

In this letter, the Siamese architecture is integrated to TDANet to learn the difference between the desired state of the target object and the state of the observed objects in the current visual observation for zero-shot tasks. Combining the target attention module and Siamese architecture, the proposed TDANet is expected to achieve a strong zero-shot capability with unseen target objects while maintaining satisfactory navigation performance with seen targets.

III. Task Definition

In the object-goal visual navigation task, the agent navigates to a target object t defined in the target class set T given only the egocentric RGB image without prior knowledge of the environment. In the zero-shot visual navigation task, the agent learns to navigate to a target object defined in the seen class S, and navigates to a target object defined in the unseen class U where $S \cap U = \emptyset$.

The initial positions of the agent and the target object are randomly selected in each episode. The agent samples its action a through a policy network π using the current RGB image I and the word embedding of the target object w_t as the input, *i.e.*, $a \sim \pi_\theta(I, w_t)$. Here, θ is the weight of the policy network. $a \in \mathcal{A} = \{$`MoveAhead`, `RotateLeft`, `RotateRight`, `LookUp`, `LookDown`, `Done`$\}$. The `MoveAhead` action moves the agent forward 0.25m. The angles of `Rotate` and `Look` actions are $45°$ and $30°$, respectively. Finally, the `Done` action represents the situation when the agent determines that it has found the target and thus ends the episode. An episode is considered a success, when the agent samples the `Done` action and the target object is `visible`. The `visible` indicates that the target object is in the agent's current RGB observation as well as within a distance of 1.5m from it.

IV. Target-Directed Attention Network

The overview of the proposed target-directed attention network (TDANet) is illustrated in Fig. 1. The observation of the agent is the RGB image captured from its monocular camera. The object detector processes the image and generates bounding boxes of objects from the current visual observation. The object detection results with corresponding word embeddings and the word embedding of the target object are inputted into the target attention (TA) module and the Siamese architecture (SA) network to learn the visual representation for the navigation policy. The target attention module learns the spatial and semantic relationships between the observed objects and the target object to select features of objects that have the most correspondence with the target object. The Siamese architecture distinguishes the difference between the observed and target states and encodes it into the visual representation, enabling the zero-shot ability of TDANet. The visual representation is sequentially passed into a feed-forward network and a long short-term memory (LSTM) network to extract deeper features and store the previous memory of navigation. Finally, the asynchronous advantage actor-critic (A3C) model [23] learns the navigation policy and controls the agent

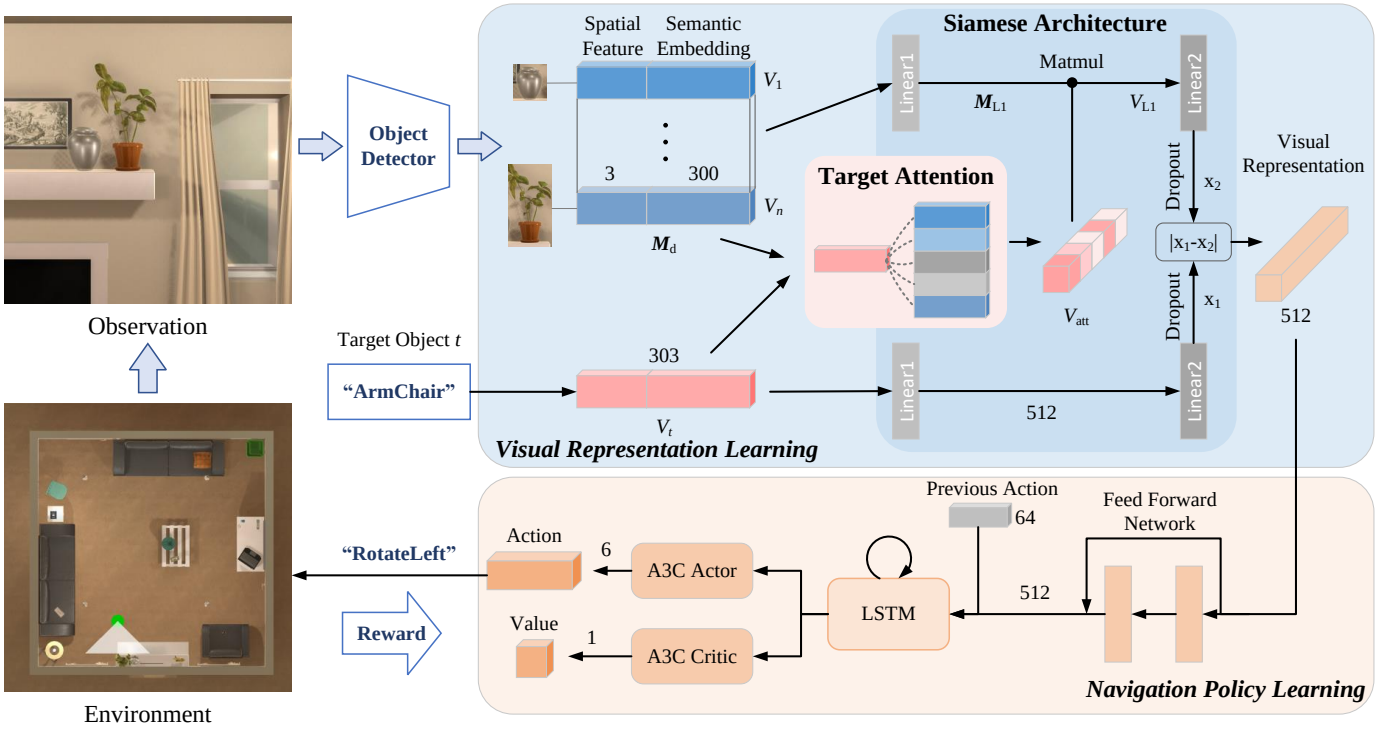

Fig. 1. Overview of TDANet. The input is the fused data of bounding boxes and word embeddings of the observed objects and the target object t. The target attention module learns the correspondence between the observed objects and the target object t and selects features of the objects most relevant to the target. The Siamese architecture compares the current state with the target state and generates the visual representation. A3C DRL model [23] is adopted to learn the navigation policy and is trained with rewards from the environment.

to navigate to the target object. The network is trained end-to-end and the reward from the environment propagates back to the TA and SA modules, which guides them to learn the meaningful visual representation containing the relationships among objects. The important modules of TDANet are detailed as follows.

A. Input of TDANet

The input of TDANet consists of two components—the detected object matrix $\boldsymbol{M}_\mathrm{d}$ and the target vector V_t. $\boldsymbol{M}_\mathrm{d}$ concatenates the bounding box data of the detected objects with the corresponding word embeddings, *i.e.*, $\boldsymbol{M}_\mathrm{d} = [V_i, i \in \{1, 2, ..., n\}] \in \mathbb{R}^{n\times 303}$, where $V_i = [x_i, y_i, S_i, w_i]$ represents the vector of spatial and semantic information of the detected object i. n is the total number of detected objects. x_i, y_i and S_i indicate the location (x, y) and the area S of the bounding box of object i in the image coordinates. w_i is the 300-dimensional word embedding of object i. $[V_i]$ represents the vertical concatenation of row vectors V_i. If there is no object detected in the current frame, $\boldsymbol{M}_\mathrm{d} \in \mathbb{R}^{1\times 303}$ is set to a zero vector. $V_\mathrm{t} = [x_\mathrm{t}, y_\mathrm{t}, S_\mathrm{t}, w_\mathrm{t}]$ is the target vector, representing the desired state of the agent to observe the target object t. x_t, y_t and S_t are set to 0.5, 0.5 and 0.25, respectively, indicating the desired location of object t is in the middle of the image with an area of a quarter of the entire image area. w_t is the word embedding of object t. Similar to [4], [6], [8], [9], we use the ground-truth bounding boxes in AI2-THOR and GloVe [24] to generate word embeddings.

B. Target Attention Module

The target attention (TA) module is proposed to learn the spatial and semantic correspondence between the detected and target objects. Taking $\boldsymbol{M}_\mathrm{d}$ and V_t as inputs, the TA module linearly maps both of them to a vector space of the same dimension and learns the relationships through matrix multiplication, generating correspondence vector V_corr, *i.e.*,

$$V_\mathrm{corr} = (V_\mathrm{t}\boldsymbol{W}_\mathrm{L} + b_\mathrm{L})\,(\boldsymbol{M}_\mathrm{d}\boldsymbol{W}_\mathrm{L} + b_\mathrm{L})^\top \tag{1}$$

Here, $\boldsymbol{W}_\mathrm{L} \in \mathbb{R}^{d_\mathrm{input}\times d_\mathrm{output}}$ and $b_\mathrm{L} \in \mathbb{R}^{n_\mathrm{rows}\times d_\mathrm{output}}$ are the trainable weight and bias of the linear layer, respectively. d_input and d_output are the input and output dimensions of the linear layer, respectively. n_rows is the number of rows of the input matrix. $V_\mathrm{corr} \in \mathbb{R}^{1\times n}$ encodes the correspondence of the detected objects with the target object. n is the number of detected objects in the current frame. The TA module then calculates the vector of the attention probability distribution on correspondence values, denoted by V_att, by applying a softmax function to V_corr, *i.e.*,

$$V_{\mathrm{att},i} = \frac{\exp(V_{\mathrm{corr},i})}{\sum_j^n \exp(V_{\mathrm{corr},j})} \tag{2}$$

where $V_{\mathrm{att},i}$ and $V_{\mathrm{corr},i}$ denote the i-th values of V_att and V_corr, respectively. n is the number of detected objects.

The vector output of the TA module $V_\mathrm{att} \in \mathbb{R}^{1\times n}$ is then multiplied by the extracted feature matrix $\boldsymbol{M}_\mathrm{L1} \in \mathbb{R}^{n\times d_\mathrm{L1}}$ to obtain V_L1, *i.e.*, $V_\mathrm{L1} = V_\mathrm{att}\boldsymbol{M}_\mathrm{L1}$. $\boldsymbol{M}_\mathrm{L1}$ represents the extracted features of the n detected objects by passing $\boldsymbol{M}_\mathrm{d}$

through layer "Linear1" whose output dimension is d_{L1}. V_{L1} represents the weighted average feature vector of $\boldsymbol{M}_{\mathrm{L1}}$ based on the attention distribution. This operation selects the features of detected objects most related to the target object.

The TA module learns the relationships between the detected objects and the target object during training, expected to help the agent focus on the most relevant objects to the target during navigation, thus improving the navigation success rate and efficiency. In addition, the design of the TA module permits an input of object detection results from an arbitrary number of observed objects. Compared to other work using the one-hot encoding [3] or an input matrix of fixed size [6], [9], the proposed TA module allows the update of the detection results when new objects are observed, enhancing the generalization ability.

C. Siamese Architecture

The Siamese neural network (SNN) is commonly used in few-shot learning tasks such as face recognition [25] and signature verification [26] where it is difficult to train every instance of the data. SNN usually contains two branches of sub-networks with identical parameters to learn the difference between two inputs. TDANet introduces the Siamese architecture (SA) design to learn the difference between the current and target states, enabling the zero-shot ability of the agent to unseen objects. The SA contains two branches of linear layers sharing the learning weights, as shown in Fig, 1. One branch extracts the feature of the detected object matrix $\boldsymbol{M}_{\mathrm{d}}$ selected by the TA module, and the other extracts the feature of the target vector V_{t}. Finally, SA calculates the absolute difference of the output vectors of the two branches with a dropout layer to avoid over-fitting.

D. Reward Function

The reward function R is designed as

$$R = \begin{cases} R_p & \text{if } p \text{ is } \texttt{visible} \\ R_t & \text{if } t \text{ is } \texttt{visible} \text{ when } \texttt{Done} \\ R_t + R_p & \text{if } t\&p \text{ are } \texttt{visible} \text{ when } \texttt{Done} \\ -0.01 & \text{otherwise} \end{cases} \tag{3}$$

Here, R_t and R_p are the target reward and the partial reward, respectively. t and p represent the target object and the parent object, respectively. When the `Done` action is sampled, the agent receives a target reward $R_t = 5$ if the target object is `visible`. We introduce R_p proposed in [6] using the parent-target relationship to help the agent learn the relationship among objects. The parent objects are the larger objects related to the target object t in a room. R_p is calculated through $R_t \cdot \Pr(t \mid p) \cdot k$. Here, k is a scaling factor set to 0.1. $\Pr(t \mid p)$ denotes the conditional probability of finding t in the neighborhood of p given the agent observing p. It is calculated based on the relative spatial distance of all the parent objects to the target [6]. The agent receives R_p when a parent object p is `visible` in the current RGB frame. In the case of observing the same p, the agent does not receive R_p again to encourage exploration. The penalization of -0.01 is used to foster a shorter path.

V. Experiments

A. Experimental Setup

We have conducted extensive experiments in the AI2-THOR embodied AI environment to train and test the proposed TDANet. The environment contains 120 near photo-realistic indoor room scenes, including 30 scenes from each of the four room types, *i.e.*, kitchen, bedroom, living room, and bathroom. Following the literature [6], [7], [9], [13], we select the first 20 rooms from each room type as the training set and the rest 10 rooms from each room type as the test set. The rooms in the test set are unseen during training. The commonly used 22 classes of objects are selected as the target set.

The agent is trained for 6000,000 episodes on the offline data from AI2-THOR v1.0.1 with a learning rate of 0.0001. During the test, 250 episodes for each room type are evaluated. The evaluation metrics include the success rate (SR) and the success weighted by path length (SPL) [27], which are commonly adopted in existing visual navigation studies [6], [7], [9], [13], [27]. SR is calculated as $\frac{1}{N}\sum_{i=1}^{N} S_i$, where N is the total number of episodes and S_i is a binary success indicator of the i-th episode. SPL is calculated as $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{L_i}{\max(L_i, e_i)}$ where e_i is the path length of the agent in the i-th episode and L_i is the optimal path length from the agent's initial state to the target object. We evaluate the performance of the trained agents in two sets of episodes where the optimal path length is greater than 1 ($L \geq 1$) as well as greater than 5 ($L \geq 5$), separately.

B. Comparison Models

Several benchmark object-goal visual navigation models are selected for comparison, detailed as follows. **Random**. The agent samples its actions following a uniform probability distribution. **Baseline** [9], [12] concatenates the ResNet feature extracted from the current RGB image with the GloVe embedding of the target object as the input. **SP** [5] utilizes the prior knowledge of scenes to train the navigation policy. **SAVN** [13] learns to adapt to new environments during both training and inference using a meta-reinforcement learning approach. **MJOLNIR** [6] introduces a novel context vector in the graph convolutional neural network to learn the hierarchical object relationship. **Li *et al.*** [7] combines hierarchical object relationship learning with meta-reinforcement learning, which achieves a state-of-the-art navigation performance in unseen scenes. **SSNet** [9] is a state-of-the-art model of zero-shot visual navigation, which uses object detection results and cosine similarity of word embeddings as inputs to reduce class-related dependency.

C. Experiments of Seen Objects in Unseen Scenes

We train 5 independent agents using TDANet in the training set with all the 22 classes of target objects and deploy them in the unseen scenes in the test set. It takes about 17 hours to train 6M episodes for each agent using three NVIDIA GeForce RTX 4090 GPUs. The average SR and SPL of TDANet in the test set are shown in Table I with comparison results from other models using the same experimental setup.

TABLE I
COMPARISON RESULTS WITH STATE-OF-THE-ART MODELS ON SEEN OBJECTS IN THE TEST SET.

Model	$L \geq 1$		$L \geq 5$	
	SR (%)	SPL (%)	SR (%)	SPL (%)
Random	11.2	5.1	1.1	0.5
Baseline [12]	35.0	10.3	25.0	10.5
SP [5]	35.4	10.9	23.8	10.7
SAVN [13]	35.7	9.3	23.9	9.4
MJOLNIR-r [6]	54.8	19.2	41.7	18.9
MJOLNIR-o [6]	65.3	21.1	50.0	20.9
SSNet [9]	63.7	22.8	42.9	21.3
Li *et al.* [7]	71.0	19.6	61.9	24.2
TDANet (ours)	**78.2** ±0.8	**30.6** ±0.8	**67.0** ±1.2	**33.4** ±0.8

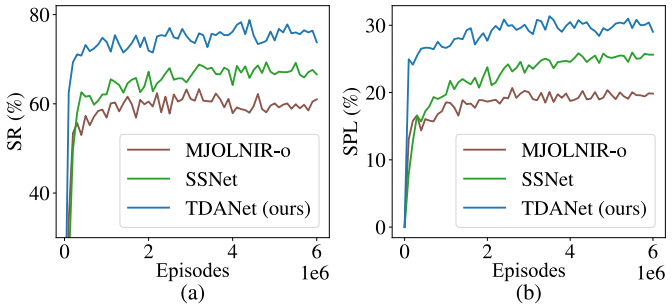

Fig. 2. The SR and SPL in the test set evaluated at each training episode of selected comparison models.

TDANet significantly outperforms all the selected models, with an increase of 7.2% in SR and 7.8% in SPL when $L \geq 1$, and an increase of 5.1% in SR and 9.2% in SPL when $L \geq 5$. It surpasses the state-of-the-art MJOLNIR and the model proposed by Li *et al.*, both of which require prior construction of a knowledge graph. The results suggest that TDANet successfully learns the spatial and semantic relationships between the target object and the observed objects during training and generalizes well to unseen scenes in the test set.

The SR and SPL in the test set evaluated at each training episode of different models are shown in Fig. 2. The SR and SPL of TDANet increase more rapidly and reach higher values than other models. Table II lists the numbers of parameters and average inference time of different models calculated using an NVIDIA GeForce RTX 3060 GPU. Although TDANet has more parameters than SSNet, the difference in the inference time between them is only 0.1 ms, which is typically negligible in the task of robot navigation.

To analyze the learned relationships using the target attention (TA) module of TDANet, we investigate the correspondence values of observed objects to the target object in the correspondence vector V_{corr} learned using Eq. (1). Figure 3 visualizes the objects of interest with the darker red color representing a higher predicted correspondence value in the current visual observation. We observe that TA consistently predicts the highest value for the target object and higher

TABLE II
THE NUMBERS OF PARAMETERS AND INFERENCE TIME OF DIFFERENT MODELS.

Model	Param.	Time (ms)
MJOLNIR-o [6]	3.42M	0.6
SSNet [9]	2.00M	0.5
TDANet(ours)	3.08M	0.6

(a) (b) (c)

Fig. 3. The visualization of the TA module. Only the bounding boxes of objects with higher correspondence values are labeled. The darker red color of a bounding box indicates a higher correspondence value. The target object is marked with the blue bounding box. The results demonstrate that the TA successfully learns the correspondence between the objects and the target.

values for other objects either spatially or semantically more related to the target object. For example, Fig. 3(a) predicts the target object `RemoteControl` is highly related to the `TableTop`, `Sofa` and `Television`. It suggests that TA pays more attention to the objects where the target is potentially found in their neighborhood, thus improving the navigation success rate and efficiency.

Figure 4 shows the comparison between the predicted trajectories of MJOLNIR [6] and TDANet. TDANet finds the target with the shortest path with a higher success rate, consistent with the experimental results presented in Table I.

D. Zero-Shot Experiments

The 22 target objects are split into seen and unseen object classes similar to [9]. TDANet is first trained to navigate to objects in the seen class in the training set and then deployed in the test set. The detection results of objects in the unseen class are removed and not inputted into the network during training so that the network does not learn any information about the unseen objects.

Table III shows the evaluation results of the comparison experiments of the zero-shot navigation. It is observed that TDANet achieves the overall best performance with a significant increase of SR and SPL both in the seen and unseen classes. In the task of seen target navigation, TDANet significantly outperforms the state-of-the-art zero-shot model SSNet by an increase of more than 23% in SR and an increase of 13% in SPL, when the target object is far from the agent's initial position ($L \geq 5$). In the task of unseen target navigation, TDANet surpasses SSNet by a large margin on SR and SPL of more than 28.1% and 13.5%, respectively. We conjecture the reason for the improved performance of TDANet over SSNet is as follows. Firstly, The input of SSNet is a fixed-size matrix only containing detection results of a set of predefined

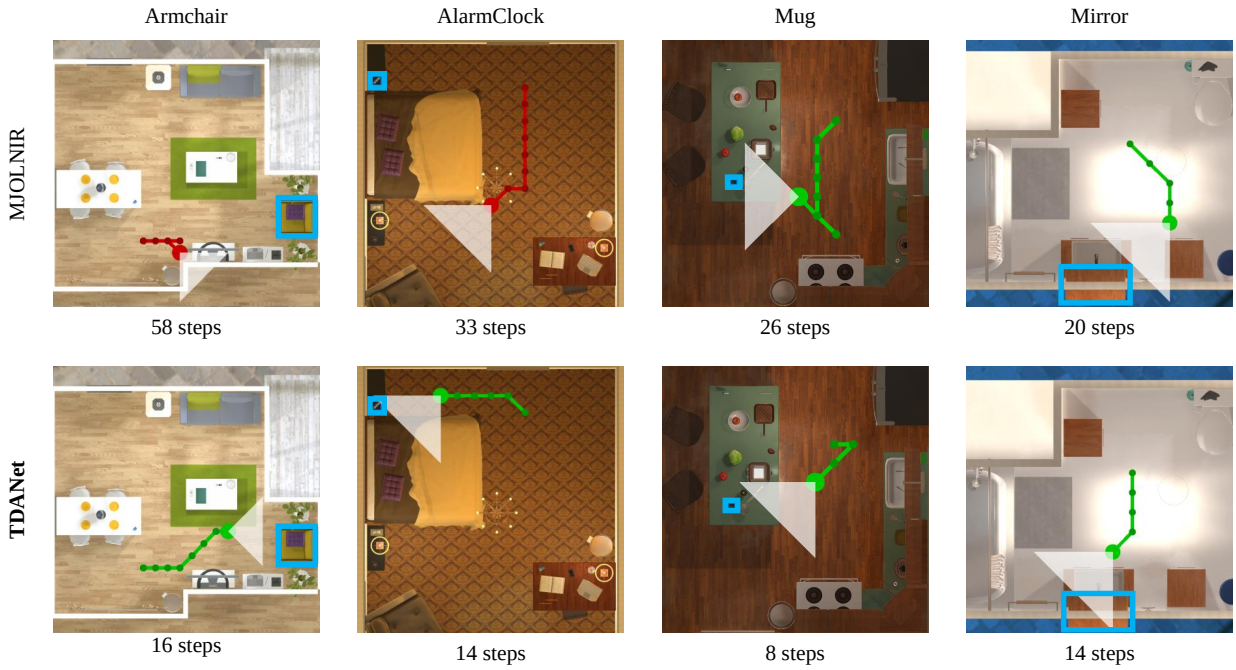

Fig. 4. The sampled trajectories of MJOLNIR [6] and our TDANet along with the number of navigation steps. Red and green trajectories represent success and failure, respectively. The white triangle indicates the field of view of the agent. The target object is marked with a blue bounding box.

TABLE III
COMPARISON RESULTS OF THE ZERO-SHOT EXPERIMENTS WITH STATE-OF-THE-ART MODELS IN THE TEST SET.

Model	Seen/ Unseen split	Unseen class				Seen class			
		$L \geq 1$		$L \geq 5$		$L \geq 1$		$L \geq 5$	
		SR (%)	SPL (%)	SR (%)	SPL (%)	SR (%)	SPL (%)	SR (%)	SPL (%)
Random	18/4	10.8	2.1	0.9	0.3	9.5	3.3	1.0	0.4
Baseline [9]		16.9	8.7	5.3	3.1	17.7	8.3	5.3	2.6
MJOLNIR [6]		20.7	7.1	10.6	4.5	51.9	16.5	33.0	14.2
SSNet [9]		28.6	9.0	12.5	5.6	59.0	19.7	38.6	18.3
TDANet (ours)		**62.5**	**25.3**	**47.4**	**23.8**	**74.7**	**29.7**	**62.9**	**32.2**
Random	14/8	8.2	3.5	0.5	0.1	8.9	3.0	0.5	0.3
Baseline [9]		14.6	4.9	4.9	2.8	30.4	9.7	11.5	5.2
MJOLNIR [6]		12.3	5.1	6.0	3.6	52.7	22.3	26.8	14.9
SSNet [9]		21.5	7.0	13.0	6.7	59.3	24.5	35.2	19.3
TDANet (ours)		**53.4**	**20.5**	**41.1**	**20.7**	**77.1**	**31.0**	**64.2**	**33.6**

objects and cannot update itself with the detection results of unseen objects. In contrast, the TA and SA design of TDANet allows the update of the object detection module with detection results of new objects so that it generalizes significantly better to the navigation task with unseen objects. Secondly, instead of directly using the cosine similarity of word embeddings as SSNet, the TA module adaptively learns the semantic relationships together with spatial relationships to learn domain-independent features. Thirdly, SSNet fuses all features of predefined objects (more than 90 categories), regardless of whether or not it appears in the current observation. In contrast, the TA module only focuses on the most related objects to the target in the current observation, which is more efficient. Fourthly, while SSNet uses cosine similarity as the class-independent data, TDANet predicts actions based on the difference between the current and target states learned by the SA module instead of the certain object class data, resulting in the improved generalization ability.

The results demonstrate that TDANet has robust generalization capabilities in the zero-shot task for unseen objects in unseen scenes while simultaneously maintaining high navigation performance with seen objects during training.

E. Ablation Study

The ablation study is conducted to evaluate the influence of the target attention (TA) module and the Siamese architecture (SA) of TDANet. For removing the TA module, we calculate the average features of all observed objects. We

TABLE IV
ABLATION STUDY FOR TDANET ON SEEN OBJECTS IN THE TEST SET.

Module		$L \geq 1$		$L \geq 5$	
TA [1]	SA [2]	SR (%)	SPL (%)	SR (%)	SPL (%)
✗	✗	50.0	13.9	36.8	14.7
✗	✓	58.9	15.8	41.6	15.1
✓	✗	76.3	28.2	62.7	30.4
✓	✓	78.8	30.6	67.7	33.4

[1] TA: Target attention
[2] SA: Siamese architecture

TABLE V
ABLATION STUDY FOR TDANET USING 18/4 SEEN/UNSEEN CLASS SPLIT OF ZERO-SHOT SETTING

Module		Seen class		Unseen class	
TA [1]	SA [2]	SR (%)	SPL (%)	SR (%)	SPL (%)
✗	✗	42.1	10.6	8.1	1.2
✗	✓	56.5	15.7	17.0	4.2
✓	✗	63.3	20.0	14.2	2.1
✓	✓	74.7	29.7	62.5	25.3

[1] TA: Target attention
[2] SA: Siamese architecture

follow the same experimental setup and evaluation metrics in Section V-C. The ablation study results on seen objects are listed in Table IV. We observe that both the TA and SA modules improve SR and SPL. In addition, TA contributes more to the improvement of navigation performance. We conjecture that TA helps the agent focus more on the spatial and semantic features of the most relevant observed objects to the target, thus improving the success rate and navigation efficiency especially when dealing with trained target objects.

To analyze the zero-shot performance of TDANet, another ablation study under the same experimental setup in Section V-D is conducted and the results of the zero-shot setting are listed in Table V-D. It is observed that the combination of the TA and SA modules significantly increases SR and SPL by more than 40% and 20% in the zero-shot task, respectively. The design of the TA and SA modules significantly improves the zero-shot ability of TDANet. The TA module selects features of the observed objects related to the target, which are inputted into the SA module to learn the difference between the target and current states. TDANet then predicts actions based on the class-independent features outputted from the SA module, thus improving the zero-shot navigation ability. In comparison, When using the SA module only, the average feature of all observed objects is calculated and inputted into the SA module. As illustrated in Fig. 5, the SA-only network is distracted by unrelated objects without the help of the TA module and predicts wrong actions when it finds the target, while TDANet predicts the right action by focusing on the objects related to the target.

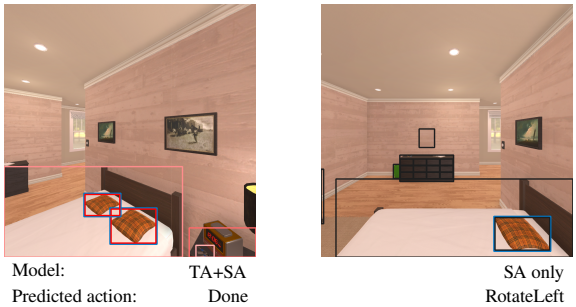

Fig. 5. The comparison of TDANet and the SA-only network for unseen object goal navigation. The target object `Pillow` is marked with the blue bounding box. (a) TDANet predicts the right action by focusing on objects related to the target. (b) Without the help of the TA module, the SA-only network is distracted by unrelated objects and predicts the wrong action.

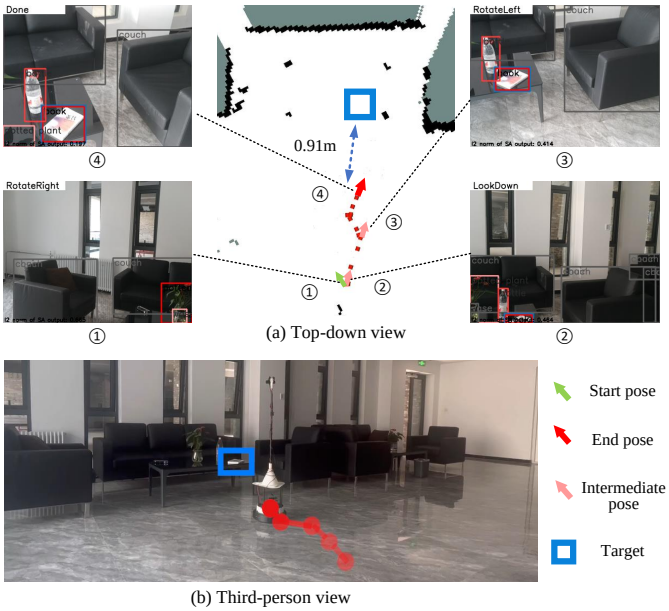

Fig. 6. A sample trajectory of the real-world deployment of TDANet to the unseen object `Book`. (a) Top-down view of the trajectory. ①-④ are sampled egocentric RGB observations of the robot marked with detected bounding boxes of YOLOv7 and the predicted action of TDANet at different poses of the trajectory. The darker red color of the bounding box represents a higher correspondence value predicted by the TA module. (b) Third-person view of the scene and the robot trajectory.

F. Real-world Deployment

A testing system using a TurtleBot4 wheeled robot is developed and an OAK-D-Pro camera is installed on the robot at a height of 1.5m above the ground to test the real-world generalization of TDANet. A servo is equipped to control the camera's rotation. Navigation only required the RGB frame of the camera. The agent trained in Section V-C is used to deploy and YOLOv7 [28] pretrained in the COCO dataset is used as the object detector of TDANet. All algorithms run in real-time on a laptop with an i7-12700H CPU and an NVIDIA GeForce RTX 3060 GPU. A sample navigation trajectory of the agent to an unseen object is visualized in Fig. 6. Videos are available in the supplementary items. TDANet successfully navigates the robot to the unseen target object `Book` by focusing on

the observed objects that are most related to the target in the current observation. For example, in Fig. 6①, `PottedPlant` in the bottom right corner of the image is predicted as the most related object and an action of `RotateRight` is predicted. In Fig. 6②, TDANet locates the target at the bottom of the image and predicts the action `LookDown`. The experimental results illustrate the satisfactory generalization capability of TDANet to the real world.

VI. Conclusion

This letter proposed the target-directed attention network (TDANet), which paid more attention to the most relevant objects to the target object in the monocular visual observation during navigation. A target attention module was designed to learn the spatial and semantic correspondence of the observed objects with the target object. The adopted Siamese architecture compared the current state to the target state, improving the generalization of TDANet. Extensive comparison experiments and ablation studies in the AI2-THOR environment were conducted, the results of which demonstrated that TDANet learned a domain-independent visual representation for navigation policy with a strong generalization ability to both unseen scenes and unseen target objects, achieving higher navigation success rate and efficiency compared to other selected state-of-the-art models. The deployment of TDANet in the real world demonstrated its generalization ability in real-world environments.

In future work, we plan to investigate the collision avoidance of TDANet in more complex simulated environments [29] for a safe application in the real world.

References

[1] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *Proc. Eur. Conf. Comput. Vision*, 2020, pp. 19–34.

[2] Y. Lyu, Y. Shi, and X. Zhang, "Improving target-driven visual navigation with attention on 3d spatial relationships," *Neural Process. Lett.*, vol. 54, pp. 3979 – 3998, 2022.

[3] H. Du, X. Yu, and L. Zheng, "VTNet: Visual transformer network for object goal navigation," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–16.

[4] R. Fukushima, K. Ota, A. Kanezaki, Y. Sasaki, and Y. Yoshiyasu, "Object memory transformer for object goal navigation," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 11 288–11 294.

[5] W. Yang, X. Wang, A. Farhadi, A. K. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–14.

[6] A. Pal, Y. Qiu, and H. Christensen, "Learning hierarchical relationships for object-goal navigation," in *Proc. Conf. Robot Learn.*, vol. 155, 2021, pp. 517–528.

[7] F.-F. Li, C. Guo, H. Zhang, and B. Luo, "Context vector-based visual mapless navigation in indoor using hierarchical semantic information and meta-learning," *Complex Intell. Syst.*, vol. 9, pp. 2031–2041, 2022.

[8] R. Druon, Y. Yoshiyasu, A. Kanezaki, and A. Watt, "Visual object search by learning spatial context," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1279–1286, 2020.

[9] Q. Zhao, L. Zhang, B. He, H. Qiao, and Z. Liu, "Zero-shot object goal visual navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 2025–2031.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–22.

[11] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, A. Kembhavi, A. K. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," 2017, *arXiv:1712.05474*.

[12] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3357–3364.

[13] 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 *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 6743–6752.

[14] B. Mayo, T. Hazan, and A. Tal, "Visual navigation with spatial attention," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2021, pp. 16 893–16 902.

[15] Q. Zhao, L. Zhang, B. He, and Z. Liu, "Semantic policy network for zero-shot object goal visual navigation," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7655–7662, 2023.

[16] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2023, pp. 23 171–23 181.

[17] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," in *Proc. Int. Conf. Mach. Learn.*, 2024. [Online]. Available: https://openreview.net/forum?id=Va7mhTVy5s

[18] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," in *Proc. Advances Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 32 340–32 352.

[19] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2022, pp. 14 809–14 818.

[20] N. Xu, W. Wang, R. Yang, M. Qin, Z. Lin, W. Song, C. Zhang, J. Gu, and C. Li, "Aligning knowledge graph with visual perception for object-goal navigation," 2024, *arXiv:2402.18892*.

[21] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, vol. 1, 2005, pp. 539–546.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[23] 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 *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1928–1937.

[24] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[25] F. Mokhayeri and E. Granger, "Video face recognition using siamese networks with block-sparsity matching," *IEEE trans. biom. behav. identity sci.*, vol. 2, no. 2, pp. 133–144, 2020.

[26] X. Wu, A. Kimura, S. Uchida, and K. Kashino, "Prewarping siamese network: Learning local representations for online signature verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 2467–2471.

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

[28] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2023, pp. 7464–7475.

[29] J. Ma, H. Dai, Y. Mu, P. Wu, H. Wang, X. Chi, Y. Fei, S. Zhang, and C. Liu, "Doze: A dataset for open-vocabulary zero-shot object navigation in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 9, pp. 7389–7396, 2024.