IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 16, 2020, accepted September 3, 2020, date of publication September 9, 2020,
date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023014

Model-Agnostic Metalearning-Based Text-Driven Visual Navigation Model for Unfamiliar Tasks

TIANFANG XUE[1,2,3,4], AND HAIBIN YU[1,2,3], (Senior Member, IEEE)
[1]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[4]University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Haibin Yu (yhb@sia.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1700200; in part by the National Natural Science Foundation of China under Grant 61803368, Grant 61533015, Grant 61972389, and Grant 61903356; in part by the China Postdoctoral Science Foundation under Grant 2019M661156; in part by the Liaoning Provincial Natural Science Foundation of China under Grant 20180540114 and Grant 20180520029; in part by the Youth Innovation Promotion Association CAS; and in part by the Independent Subject of State Key Laboratory of Robotics.

ABSTRACT As vision and language processing techniques have made great progress, mapless-visual navigation is occupying uppermost position in domestic robot field. However, most current end-to-end navigation models tend to be strictly trained and tested on identical datasets with stationary structure, which leads to great performance degradation when dealing with unseen targets and environments. Since the targets of same category could possess quite diverse features, generalization ability of these models is also limited by their visualized task description. In this article we propose a model-agnostic metalearning based text-driven visual navigation model to achieve generalization to untrained tasks. Based on meta-reinforcement learning approach, the agent is capable of accumulating navigation experience from existing targets and environments. When applied to finding a new object or exploring in a new scene, the agent will quickly learn how to fulfill this unfamiliar task through relatively few recursive trials. To improve learning efficiency and accuracy, we introduce fully convolutional instance-aware semantic segmentation and Word2vec into our DRL network to respectively extract visual and semantic features according to object class, creating more direct and concise linkage between targets and their surroundings. Several experiments have been conducted on realistic dataset Matterport3D to evaluate its target-driven navigation performance and generalization ability. The results demonstrate that our adaptive navigation model could navigate to text-defined targets and achieve fast adaption to untrained tasks, outperforming other state-of-the-art navigation approaches.

INDEX TERMS Mapless-visual navigation, semantic segmentation, text-driven, model-agnostic meta-learning.

I. INTRODUCTION

Nowadays substantial researches have been carried out in mapless robot navigation field. Agents governed by goal-based tasks are specifically designed to navigate only depending on visual information with little prior knowledge of the environment, resulting in less system cost and power consumption. In addition to image processing, mapless visual navigation requires agent to interact with the environment efficiently, where deep reinforcement learning method has been adopted. In recent studies, DQN [1]and A3C [2], considered as the most representative RL algorithms, are widely used in navigation field to realize interactive process. Based on such end-to-end learning mechanism, navigation model is enabled to eliminate errors accumulated from traditional engineering projects, such as extracting visual features, making map, identifying object location and planning path. The performance of the whole system can be greatly improved and maintained.

However, a great challenge is still existing in recent DRL-based navigation studies [3]. Since DRL models are considered to be black-box models with unalterable structure, they have made quite poor performance in generalization. As a model is well trained for a specific task, it can be hardly implemented to other targets or environments. Although some works have been proposed on generalizing

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Wang.

pre-trained models to unfamiliar tasks, such as target-driven network [4], dueling network [5], context grid [6] and multi-view representation learning [7], these methods fail to make full use of previous experience and guarantee their stability when dealing with novel experiments. To tackle this challenge, rather than setting up multi-tasking network or other similar approaches to improve compatibility, we introduce meta-learning mechanism and enable our navigation model to integrate its prior experience with new cognition obtained from the current task. After exploring appropriate amount of episodes in training environments, the agent is allowed to learn and adapt in untrained environment as parameters of its model altered. Avoiding over-fitting to the new task, This adaption requires no further explicit supervision but a few interaction with novel surroundings.

In this study, a novel text-driven visual navigation model has been proposed to accomplish untrained navigation tasks in novel environment. In our model, Images observed by agent and text-defined target are considered as inputs of this DRL network, with the actor-critic network [8] outputting sequences of action. To construct more direct and convenient connection between target and current states of the agent, we introduce Fully Convolutional Instance-aware Semantic Segmentation(FCIS) [9] and Word2vec [10] as preprocessing tools so as to encode visual observation and text-defined goal into vectors with semantic relatedness. After preprocessing phase, visual features and semantic features are embedded corresponding to a siamese deep reinforcement learning network [11], providing decision-making basis. Benefitting from such network, our self-adapting learning approach, which is derived from Model-Agnostic Meta-Learning (MAML) [12], [13], can promote initial parameters rapidly reaching values which are most susceptible to variation of tasks. When dealing with new tasks these parameters proceed to converge by exploring in the new environment, until the model finally achieves adaption. Unlike conventional DRL methods, we remove the rigid boundary between training phase and testing phase: agent can learn experience via interaction with current scene and keeps modifying its network parameters during the whole process.

The proposed model has been performed on Matterport3D dataset [14], which includes a great many RGB images of indoor scenes and has been widely applied for both theoretic and practical engineering researches. Several experiments have been designed to evaluate the target-driven navigation performance and generalization performance of this model. We also compare our model with other current approaches to assess the limit of its capability.

II. RELATED WORK

A. DRL NAVIGATION

Recently learning-based navigation has become a hot topic in the visual navigation field. Unlike traditional map-based navigation methods [15]–[17] or SLAM-based techniques [18]–[20], deep reinforcement learning method doesn't require a global map of current environment to support navigation decisions. The combination of visual information and DRL mechanism can implicitly accomplish engineering projects such as localization, mapping, and path planning in end-to-end manner, with environment information embedded in network parameters. Ye-Hoon Kim *et al.* [21] propose an end-to-end navigation method to extract visual features directly from images by the camera, which greatly reduces the power consumption and computational time. The experiment is performed in an office scene with a simplified DRL model, achieving satisfying results. In order to acquire reliable sequence of actions towards goals, Saurabh Gupta *et al.* [22] propose Cognitive Mapper and Planner, a novel neural architecture for robot navigation. This architecture maintains a metric belief of the world and crucially utilizes a hierarchical version of value iteration to plan paths to distant goals. Zhu *et al.* [4] address target-driven navigation problem using a novel deep siamese actor-critic network, which takes target image as input in addition to scene image, providing compatibility for diverse targets.

B. VISION AND LANGUAGE

Since the targets of same category can possess quite diverse features, many studies have taken context vocabulary or instructions into consideration to define goals in visual navigation tasks. Dipendra Misra *et al.* [23] proposes a fusion model which maps raw visual observations and text input to actions for instruction execution. All the images embedded with texts are processed by LSTM and CNN to jointly reason about actions in an 2D block environment. Wu *et al.* [24] presents embodied agents in a simple maze world and task them to complete a series of instructions. As semantic segmentation is critical to understanding the contents in images, various convolutional neural networks have been brought into field to perform pixel-wise segmentation. Noha Radwan *et al.* [25] presents a vision-based navigation control strategy for a wheeled robot traveling outside, with input images segmented according to object category to generate moving trajectory. Our work utilize FCIS network and Word2vec model to construct more efficient connection between targets and environmental observation.

C. META LEARNING

More recently, various meta-learning mechanisms have become much more popular as depending on which learning models can solve new learning tasks using only a small number of training samples. Finn *et al.* [12], [13] introduces Model Agnostic Meta Learning (MAML) which utilizes stochastic gradient descent optimization to achieve fast adaption to novel tasks. This approach can be construed as learning a good parameter initialization to make sure the model works well with only a few gradient updated. Gupta *et al.* [26] puts forward a meta-learning approach to augment the decision policy with structured noise, by which the agent is urged to adapt after a few episodes with variability limited.

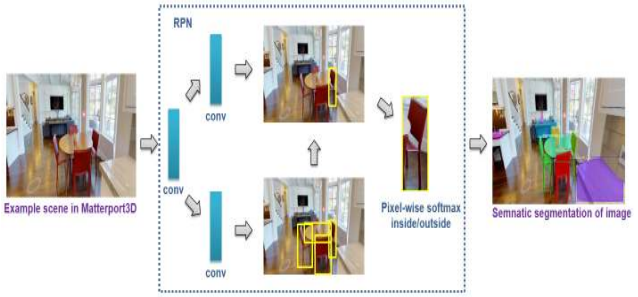

FIGURE 1. The core running procedure of FCIS network. ROIs(region of interest) are generated by RPN(Region Proposal Network) to achieve pixel-wise classification.

However, few meta-learning approaches have been applied in visual navigation field due to great computational cost from repetitive exploration. Bengio *et al.* [27] proposes a Memory-based Parameterized Skills Learning (MPSL) model for mapless visual navigation. These parameterized skills can be learnt to instruct agent behaviour for untrained tasks, facilitating task-domain generalization. Unlike this work, our model adopts Model Agnostic Meta Learning method to maintain good performance in unfamiliar experiments, realizing scene-domain generalization.

III. TEXT-DRIVEN VISUAL NAVIGATION MODEL

In this section, we will formally give a thorough introduction of our adaptive text-driven navigation model. The primary goal of this work is to develop an end-to-end visual navigation model which can navigate to designative destination based on text-defined targets and visual observation. On this foundation, this study provides new insights into improving generalization ability of DRL model in navigation field. Accoding to MAML-based training mechanism our proposed model holds a significant advantage that it not only accumulates experience from training data, but also further learns local knowledge of novel tasks. After a few exploration of novel scenes, agent can achieve fast adaption by accurately accomplishing navigation tasks with great chance. The problem formulation, network structure and adaptive learning method of this model are thoroughly illuminated in the following parts.

A. PROBLEM FORMULATION

As the objective of our work is to obtain the shortest path on which agent moves from current location to its target, interactive process is considered as partially observable markov decision process(POMDP) [28], which can be formulated as a tuple (O, A, D, R). Observations $O=\{O_T, O_V\}$, including text-defined target O_T and observed images of current state O_V, is reconfigured as the input of DRL model to create compact connection between states and goals. According to decision policy, agent navigates with sequence of actions $A=\{a_1, a_2, \ldots, a_n\}$, where a presents action space, containing 3 discretized actions: moving forward,turning the camera by 30 degrees in left/right direction. Since in Matterport3D environment agent can consistently travel in the entire house rather than teleport from one room to another, a set of reachable viewpoints $P_t + 1 \subseteq V$ are retrieved to choose next view point $v_t + 1 \in P_t + 1$ to move. To ascertain $P_t + 1$, a weighted undirected graph of viewpoints in the scene could be constructed as $G =< V, E >$. In this case the next viewpoint is given as:

$P_t + 1 = \{v_t\} \bigcup \{v_i \in V | < v_t, v_i > \in E \bigwedge v_i \in C_t\}$, where v_t represents current viewpoint and C_t represents camera view scale.

In order to minimize the trajectory length to the destination, the reward $R : O \rightarrow R$ is designed as follows: if an action is taken, agent obtains reward -0.1; if agent reaches the target, reward 10 is received. At every time step t, agent instantly selects an action a from the action set A. The exploring process terminates once agent reaches its target, or a maximum number of steps have been finished. Considering generalization tests involved, we define a series of scenes $S=\{S_1, S_2, \ldots S_k\}$ and target object class $G=\{G_1, G_2, \ldots G_m\}$. Each task is denoted by τ by such tuple $\tau=(S,G)$, with groups of scenes disjointed for the training tasks $\Gamma train$ and the testing tasks $\Gamma test$. The action-value function Q could be learnt across training and testing tasks, with network parameters continuously updated, until the model generalizes to the final task.

B. NETWORK STRUCTURE

Our network structure is established as a deep reinforcement learning framework for visual navigation as figure 2 illustrating, similar to the target-driven model Zhu *et al.* [4] proposed. Unlike Zhu's work, in this article, we put forward an attached image-processing step FCIS to convert current observation input O_V into semantic segmentation input O_S, where each pixel has been assigned with a one-hot semantic class id. We also take word vectors looked up through Word2vec as the substitute for target image to be consistent with semantic segmentation vectors of observed image. Compared to previous embedded states, these preprocessed image and text can be more directly, concisely combined in lower dimensional field. Details of proposed network are described as follows.

1) ResNet50

For the visual feature extraction, we use the same ResNet50 [29] network adopted in [4] to process RGB frames received as the observation stream. This network gets access to the model with the last FC layers removed to ensure necessary information retained. All the parameters of the network keep frozen during both training and testing phase as they are pre-trained on ImageNet. After processing four previous frames of each state, it outputs an 8192-d feature as the description of current visual observation. These visual features are then passed to a fully-connected(FC) layer with ReLU activation and finally a 512-d feature is obtained.

2) FCIS NETWORK

Fully Convolutional Instance-aware Semantic Segmentation(FCIS)network is introduced for instance-aware semantic

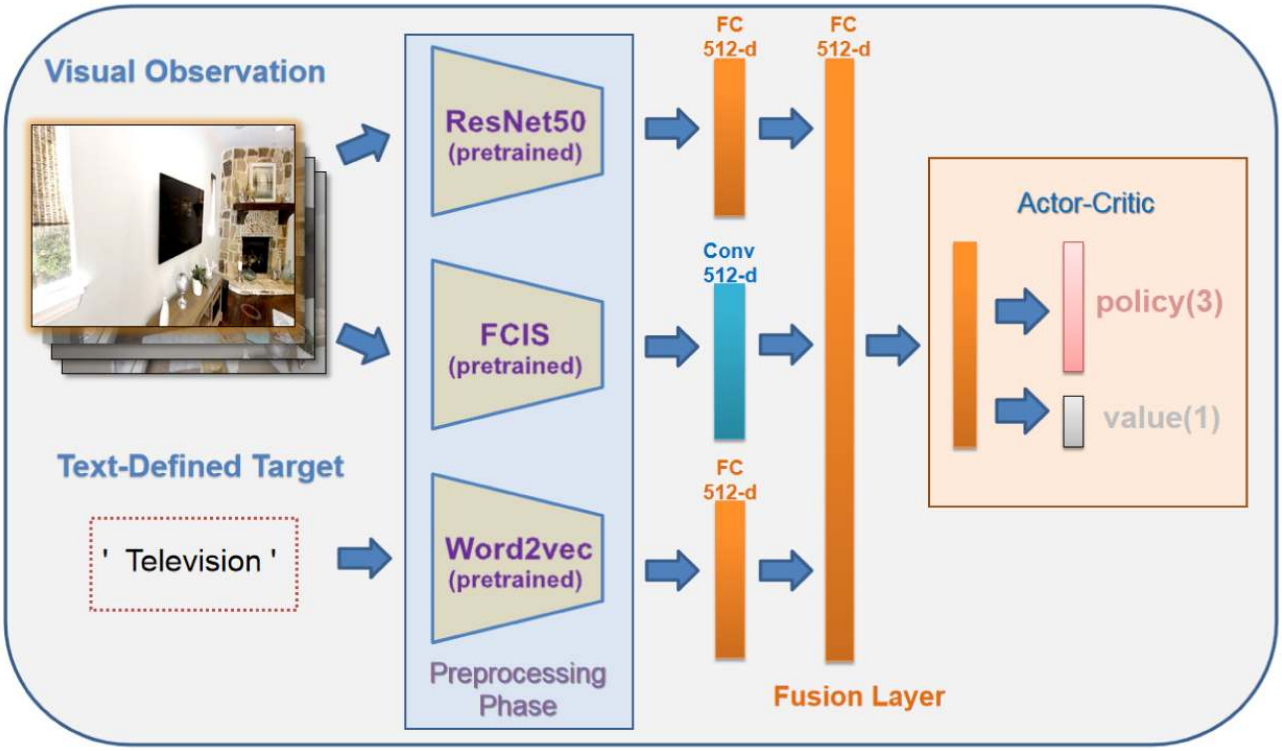

FIGURE 2. Our DRL-based network architecture. Compared to [4] we use FCIS and Word2vec model to extract semantic features and establish more efficient connection between goal and environment.

segmentation, by training a classifier to predict each pixel's likelihood score of *the pixel belongs to some object category*. Figure 1 shows the brief phase of semantic segmentation process. As different semantics may presented by the same pixel, in the FCIS, a large number of ROIs (Region of Interest) are assembled to the image to produce pixel-wise score maps. For each pixel in a ROI, it starts by detection: if it attaches to an object bounding box or not, and then examine whether it is surrounded by an object instance's boundary. Two classifies are trained as two 1×1 conv layers to obtain two sets of scores, fusing into position-sensitive inside/outside score maps to perform object segmentation and classification.

In our work, FCIS network is also pre-trained with parameters remaining unchanged. When the agent arrives to a new viewpoint, four previous frames are delivered into the FCIS. After filtered by non-maximum suppression (NMS) with an intersection-over-union (IoU, 0.3 by default), the remaining ROIs calculate their foreground masks by averaging scores of each map and weighted by classification scores, assigning one-hot semantic class id to each pixel. The output of FCIS network is fed into 4-layer convolutional net to acquire a 512-d feature vector for further processing with target encoding feature.

3) TARGET ENCODING

Compared to other image-depending navigation models such as [30], We choose natural language information to define the target instead of visual information to construct semantic relation between observation and goals. Word2vec [10] model has been implemented as another preprocessing part parallelled with FCIS, converting texts that define target objects into specific word vectors. Unlike one-hot code, these vectors are trained by context with semantic relatedness encoded. We adopt Spacy toolkit to extract embedding word vector, receiving 300-d feature per word class. As figure 2 shown, these features are then sent into a FC layer that outputs 512-d feature consistent with the output of ResNet50 and FCIS network.

4) ACTOR-CRITIC NETWORK

As the fusion layer generates the 512-d joint representation from 1536-d concatenated embedding of visual and semantic features (figure 2), Such combination of visual and textual modalities is then transferred into an actor-critic network which contains two FC layers, exporting three policy outputs and one value output. The Advantage Actor-Critic mechanism(A3C) [31] is adopted to run models in a multi-threading manner. All the gradients are back-propagated from the actor-critic network's outputs back to the upper layers, trained with a shared RMSProp optimizer of learning rate 7×10^{-4}.

C. MAML BASED LEARNING METHOD

With regard to DRL-based navigation problem, DRL model trained with specific tasks provides poor performance while

Algorithm 1 Adaptive Learning: Meta-Training Phase

Require: α, β: step hyperparameters
Require: N: termination hyperparameters
Randomly initialize θ
$n \Leftarrow 0$
while $n \neq N$ **do**
  Sample batch of tasks $\tau_i \varepsilon \Gamma_{train}$
  for all τ_i **do**
    Sample K trajectories $D = x_1, a_1, \ldots, x_m$ using f_θ in τ_i
    Evaluate $\nabla_\theta \ell_{\tau_i}(f_\theta)$ using Equation (2)
    Compute adapted parameters with gradient descent: $\theta' = \theta - \alpha \nabla_\theta \ell_{\tau_i}(f_\theta)$
    Sample trajectories $D_i' = x_1, a_1, \ldots, x_m$ using f_θ' in τ_i
  end for
  Update $\theta \Leftarrow \theta - \beta \nabla_\theta \Sigma_{\tau_i} \ell_{\tau_i}(f_\theta')$ using Equation (2)
end while

Algorithm 2 Adaptive Learning: Meta-Adapting Phase

for min-batch of tasks $\tau_j \varepsilon \Gamma_{test}$ **do**
  $\theta'' \Leftarrow \theta$
  while not converged **do**
    Sample trajectories $D'' = x_1, a_1, \ldots, x_m$ using f_θ'' in τ_j
    Evaluate $\nabla_\theta \ell_{\tau_j}(f_\theta'')$ using Equation (2)
    Update $\theta \Leftarrow \theta - \alpha \nabla_\theta'' \ell_{\tau_j}(f_\theta'')$
  end while
end for

implemented by new settings such as unfamiliar targets or environments. Gradually it becomes essential for DRL-based navigation models improving their compatibility, by means of generalizing to various kinds of tasks. Unlike scene-specific layers proposed by Zhu and parameterized skills extracted by Liu, the foundation of our learning method lies in recent meta-learning algorithm, where model trained by meta-learner can receive new experience from untrained tasks. A thorough explanation of our MAML-based learning procedure is performed in the following part.

1) MAML LEARNING

The meta-learning approach we relied on is based on the MAML algorithm [32]. MAML defines that each task $\tau \in \Gamma train$ is allocated with meta-training dataset Dtr and meta-validation dataset $Dval$. Considering image classification problem, MAML model aims to assign image class labels to each image in $Dval$ according to training examples of each class in Dtr, and get tested by unfamiliar tasks in $\Gamma test$. The learning objective function of the MAML is presented as:

$$min \sum_{\tau \in \Gamma train} \ell(\theta - \alpha \nabla_\theta \ell(\theta, Dtr), Dval) \tag{1}$$

where ℓ is loss function of network parameters θ. The main purpose is to learn parameters θ that offers great initialization for fast adaptation to untrained datasets. In our work, we put forward a navigation-specific self-adaptive learning mechanism derivated out of MAML learning.

2) ADAPTIVE LEARNING APPROACH FOR VISUAL NAVIGATION

In this article we propose a self-adaptive learning approach to learn proper network parameters that make rapid progress in generalizing to novel scenes without overfitting, such that slight changes in the parameters will generate great modification on the loss function along the direction of the gradient of that loss. Since traditional MAML technique is basically applied in image classification field, our method is designed as a modified version of MAML with an approapriate integration of screening training data and calculating related loss, which guarantees the sufficiently similar distribution of training and testing tasks. The whole running procedure can be divided to two phases: meta-training phase and meta-adapting phase.

In the meta-training phase as Algorithm 1 outlines, we aim to learn a primary model presented by parametrized function f_θ with parameters θ and a loss function of f_ϕ with step-size hyper-parameters α, β, N. After collecting batches of tasks τ_i from training datasets, specifically accomplished by randomly selecting navigation tasks in numerous scenes of the same kind, we sample K trajectories D_i using f_θ in τ_i, as the decision policy mapping from states X_t to actions a_t at each timestep t. Each RL task involves transition distribution $q_i(X_{t+1}|X_t, a_t)$ and the loss function ℓ_{τ_i} corresponding to the reward R, which takes the form:

$$\ell_{\tau_i}(f_\phi) = -E_{x_t, a_t\, f_\phi, q\tau_i}[\sum_{t=1} R_i(x_t, a_t)] \tag{2}$$

Then the adapted parameters θ' computed with gradient descent are deployed separately to sample new trajectories D_i'. With all τ_i processed, our primary adaptive model can be configured as parameters θ updated shown in line.

In the meta-adapting phase, Algorithm 2 shows when testing dataset(untrained task) appears, mini-batch of tasks $\tau_j \in \Gamma test$ are sampled, while agent performs several exploration episodes by choosing actions according to the primary model. As parameters θ further updated to θ'', our model achieves adaption in the novel environment. Generally the core concept of this work is managing K rollouts from f_θ, tasks τ_i and related rewards $R_i(x_t, a_t)$ as prior knowledge for fast adaption on untrained tasks τ_j.

Due to unknown dynamics, the expected reward is generally not differentiable, leading to policy gradient methods for estimation of model optimization. It is worthwhile mentioning that in the meta-training phase, the value of hyper-parameter N plays an important role for model modification since it decides whether the primary trained model acquires adequate prior knowledge without overfitting to training datasets. However, hand-crafted N according to experience proves to be rigid and imprecise, highly depending on preliminary work. To solve this problem, we design a

FIGURE 3. A. Agent performs adaptive phase of an navigation task (navigate to sink) in the entire house constructed by Matterport3D. The initial position is in the nearby corridor yet agent can only observe walls and doors, exploring all the rooms attached. The adaptive trajectory appears to be shambolic. b. Agent performs adaptive phase in the detached area–bedroom(red circle). With relevant visual information observed, agent can navigate to the target with less redundant moves as the adaptive trajectory shows.

combined optimization method to obtain appropriate value of hyper-parameters with details illustrated in experiment section.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENT SETUP

We configure our model into realistic scenes from Matterport3D dataset [14] to perform navigation tasks and compare its performance with other works. Matterport3D environment consists of 10800 panoramic views from 194400 RGB images in 90 scenes with 7189 paths sampled from its navigation graphs. Compared to common synthetic datasets utilized by other works, Matterport3D shows more complex scenes with multiple objects and surfaces, presenting many challenges including occlusion, scale variations, lighting variations, etc. According to dataset scale, navigation tasks (targets in scene) are selected as follows: (1) bedroom: bed, lamp and plant. (2) bathroom: toilet, sink and shower. (3) kitchen: microwave, fridge and bowl. (4) living room: television, sofa and table. For each scene type 5 scenes are chosen for training and 2 scenes for testing. We consider the action space A=moving ahead, rotating left, rotating right, while the horizontal rotation achieves in increments of 30 degrees. A navigation episode is supposed to completed if the target instance described by text input is within the field of view and the agent arrives at its nearest viewpoint, or it has taken 10000 actions failing to find target.

While agents running in the realistic environments constructed by Matterport3D dataset, an severe issue has appeared. Each house included in the dataset is greatly oversize for our navigation tasks and agents may be trapped in a task-irrelevant area or just confusingly wandering around as their sights are covered with walls, doors and branching corridors which can hardly provide agents with supports for navigation decisions (Fig.3 a). In most cases agents may exploring almost every viewpoints not only in the training phase bus also in the adaption phase, resulting in extremely low efficiency for task accomplishment. Here we split the houses in the dataset into specific areas according to scene types(bedroom/bathroom/kitchen/living room), for example, the bedroom area circled in the vertical view (Fig.3 b), to offer agents more opportunities to capture valuable observation information rather than interference factors.

For fair comparison with other baseline models, we select metrics presented by [25] which are well adopted by other navigation algorithms. The Success Rate (SR) is defined as:

$$SR = 1/N \sum_{t=1}^{n} S_i \tag{3}$$

While the Success weighted by Path Length (SPL) is defined as:

$$SPL = 1/N \sum_{t=1}^{n} S_i l_i / max(l_i, e_i) \tag{4}$$

where N is the total amount of running episodes. S_i is designed as a binary vector indicating the success of the i-th episode(if the i-th episode is a success, $S_i = 1$; else 0). l_i is the length of shortest path between the initial position and any instance of the target object and e_i is the length of current episode. All the lengths in these metircs are considered as the number of the actions taken. The length of shortest path is at least 5($l >= 5$). Considering our model demanding the meta-adapting phase for achieving maturity, these two metrics start to be calculated after 100 episodes in the meta-adapting phase.

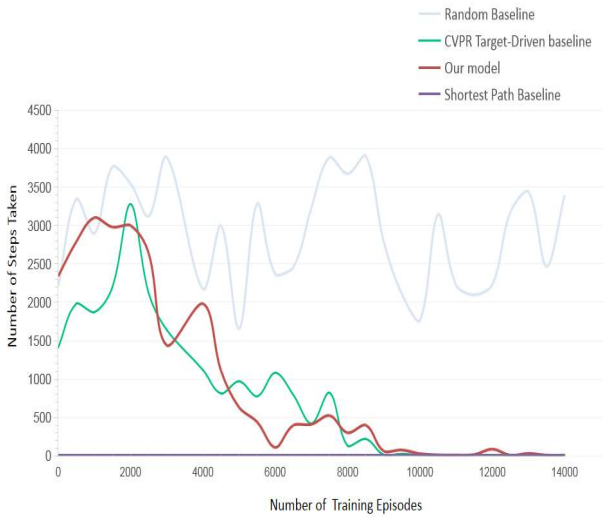

FIGURE 4. Convergence curves of all competing methods based on target-driven navigation tasks.

B. TARGET-DRIVEN NAVIGATION PERFORMANCE

We first evaluate the basic target-driven navigation performance by performing navigation tasks with other baselines, to ensure that the basic navigation accuracy has no negative influence on generalization ability. In this case our model only retains its network while disabling MAML-based adaptive learning phase. As shown in figure 4, the convergence curves of proposed model and baselines are presented, which illustrates the episode-depending change of action sequence guiding to find specific target. In this graph the x-axis indicates the amount of training episodes and y-axis indicates the number of actions taken. We compare our proposed model with three other models:

Random: Agent randomly selects an action based on a uniform distribution at each timestep.

Shortest Path: Agent is implemented with scene map and A* algorithm [33] and designed to navigate along the shortest path.

CVPR Target-Driven baseline: Model's architecture is similar to ours, but the goal and current state are illuminated by images.

Under the benchmark of Shortest Path, some text-defined targets cannot be successfully navigated due to the randomness of initial state of agent. It remains challenging for learning method to make the mean length consistent with the shortest path. However, our model has still given a quite great performance just like CVPR Target-Driven baseline over Random in average number of taken actions, hence it can be well applied to familiar goals and environments. It can observed that the convergence of our optimal solution is reached after $10k$ training episodes, demonstrating its reliability in target-driven navigation.

C. GENERALIZATION PERFORMANCE

To achieve fast adaption to unfamiliar goals or scenes in the houses constructed by Matterport3D, we train and test the navigation model according to our proposed self-adaptive learning method as Section III.C described. In the meta-training phase we randomly select 5 navigation tasks of each scene type (bedroom/kitchen/livingroom/bathroom)to compose task set $\tau_1 \sim \tau_4$. From each task set 20 trajectories $D_1 \sim D_{20}$ are sampled to compute the loss function $\ell_{\tau_1} \sim \ell_{\tau_4}$ and the parameters θ of meta-trained model are further obtained through N-based(500) iterative processes. In the meta-adapting phase testing tasks can be divided into three categories due to different scenarios:

(1) **navigating to untrained target in seen environment**;
(2) **navigating to trained target in unseen environment**;
(3) **navigating to untrained target in unseen environment**.

The meta-trained model constantly explores with novel goals or running environments allocated, until parameter θ finally converged to optimal value θ''.

Figure 5 shows an example of the learning curves in the meta-adapting phase. In this case the testing task is set as scenario 1: *navigating to untrained target in seen environment*. In the meta-training phase the agent navigates in 5 room instances, for example bedroom 01 $\sim$ 05, of each scene type. For each scene type the agent's goal remains the same. In the meta-adapting phase the agent needs to find the same target but in two unfamiliar room instances. The result shows that our model achieves primary convergence within average 50k exploring steps on trained navigation goals in all unfamiliar bedroom/kitchen/living room environments. Such performance is exceedingly superior to that of blank model, which completely retrains the model in the novel scenes with average $300k$ - $600k$ steps to converge. Hence, benefitting from previous learnt experience, our proposed model can be efficiently generalized to find the same target in new scenes. However, in neither new bathroom scenes model converges within a million steps. The cause of such failure may lies in model partial overfitting to the training scenes or the significant spatial distinction existed between training and testing environment. It may implicitly reveals that the generalization performance of meta-learning based model can be constrained by differentiated characteristics of training/testing datasets, such as room layout, illumination condition or even initial position of each episode.

We further evaluate our model by performing navigation tasks in all three scenarios with other state-of-art models. Besides from **Random** and **CVPR Target-driven Baseline** mentioned in Section IV.B, several effective models are supplemented into experiment for comparison. All the new testing models are described as follows:

MPSL: In this model agent learns parameterized skills to instruct agent behaviour for untrained tasks, facilitating task-domain generalization [27].

BRM: Model takes the form of a probabilistic relation graph over semantic entities, producing sub-goals and a goal-conditioned locomotion module for control [34].

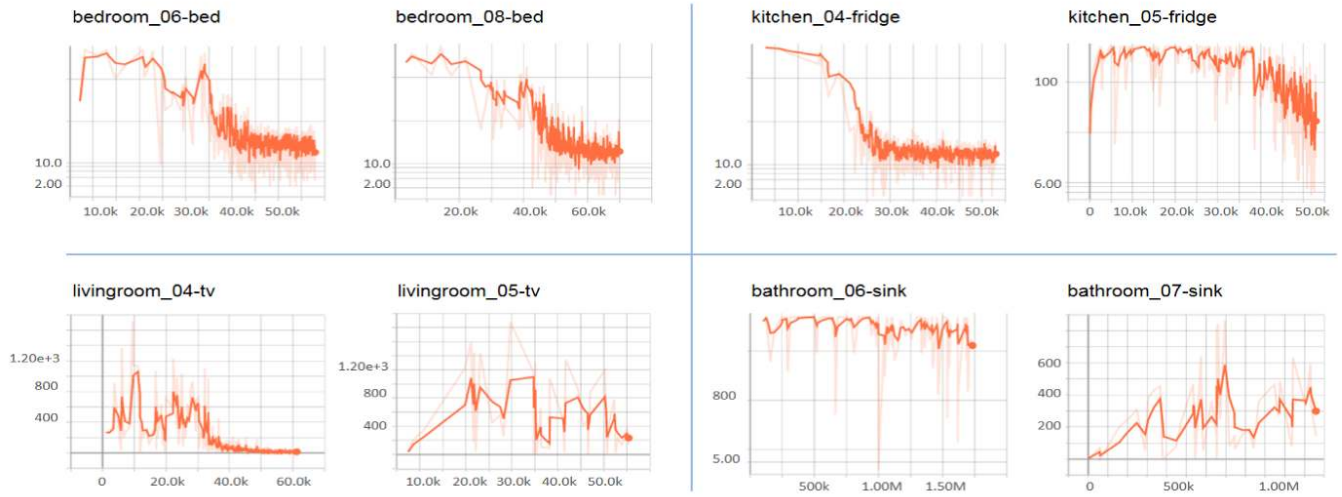

FIGURE 5. For each scene type we use 5 scenes for training and 2 scenes for testing, with the navigation target fixed. Here shows the steps-depending learning curves of testing phase in new scenes of four types. X-axis indicates the number of exploring steps taken; Y-axis indicates mean trajectory length of current model.

TABLE 1. Comparison results of SPL and success rate for all models.

Scenario	Method	Scene Instance							
		bedroom		kitchen		livingroom		bathroom	
		SPL	Suc.rate	SPL	Suc.rate	SPL	Suc.rate	SPL	Suc.rate
Untrained target in seen environment	Random	1.13	6.67	0.87	4.13	1.45	9.97	0.71	4.24
	CVPR	3.5	7.34	2.32	10.2	2.78	12.4	3.33	9.87
	MPSL	8.56	17.5	10.2	19.8	10.1	25.3	6.23	18.9
	BRM	10.3	19.9	14.6	27.1	13.1	22.2	8.97	15.4
	GCN	15.2	24.8	16.7	32.4	18.3	34.9	15.8	29.8
	Ours(vis)	15.7	23.5	11.4	27.9	8.85	17.5	10.4	24.3
	Ours(oh)	6.77	11.5	7.87	17.3	4.2	8.71	6.42	9.88
	Ours(loc)	**17.1**	**41.3**	16.9	**35.7**	**19.4**	**39.9**	**18.4**	**36.4**
	Ours(glo)	14.5	27.4	**17.7**	32.6	8.76	17.1	16.9	24.5
Trained target in unseen environment	CVPR	4.14	8.76	1.79	5.56	5.13	9.39	3.14	8.96
	MPSL	6.42	10.7	2.32	6.11	7.45	12.1	7.21	13.2
	BRM	5.56	11.2	4.43	7.80	8.91	18.7	3.22	10.4
	GCN	20.1	44.5	15.4	29.1	16.2	31.2	10.3	26.4
	Ours(vis)	17.2	29.9	13.1	22.2	17.3	34.6	9.94	23.3
	Ours(oh)	11.7	25.1	18.8	40.6	9.51	17.4	11.4	24.3
	Ours(loc)	**20.8**	**50.3**	18.9	35.8	**19.1**	**42.1**	**16.9**	**33.2**
	Ours(glo)	14.5	28.9	**19.7**	**42.1**	9.69	18.9	10.2	22.8
Untrained target in unseen environment	CVPR	5.34	16.35	3.23	13.44	8.54	16.7	2.24	8.98
	MPSL	6.93	12.1	3.34	14.5	5.28	11.4	3.76	10.5
	BRM	6.55	13.4	7.12	18.1	8.87	17.2	4.47	11.4
	GCN	10.5	21.2	14.3	33.8	9.8	13.6	9.2	18.9
	Ours(vis)	8.4	14.6	17.3	27.5	10.4	29.9	9.4	16.8
	Ours(oh)	3.65	14.1	9.32	18.7	8.2	19.8	7.43	17.2
	Ours(loc)	13.4	28.4	**19.5**	**34.1**	**13.6**	**31.9**	**10.7**	**24.4**
	Ours(glo)	**18.1**	**37.3**	10.2	20.6	11.2	29.7	8.14	18.3

GCN: Agent uses graph convolutional networks for incorporating the prior knowledge to predict the actions, achieving improvement in generalization to unseen scenes [35].

Our(vis): Our(vis) is the our model using images for indicating observation and target. All the images are processed by ResNet50 [36] instead of FCIS and Word2vec to acquire visual features.

Our(oh): Our(oh) is the our model using not word vectors received form Word2vec, but one-hot code [37] to represent navigation target. Other network structure remains the same.

Our(loc): Our(loc) corresponds to our model presented in this article, but trained and tested by room instances of one specific scene type.

Our(glo): Our(glo) is the our model presented in this article, trained and tested by room instances of all four scene types.

The navigation performances in all three scenarios of these testing models are summarized in Table 1 in terms of SPL and Success Rate(Suc.rate). It can be seen that the best results are received from our integral models(Our(loc)&Our(glo)),

which outperform the current state-of-the-art with large margins. Most notably, the success rate of Our(loc) has risen to average 30% to 40%, almost 25% increase than GCN. The reason for this can be explained by the limitation of structural design and operating mechanism in contrast models. Compared to our approach, Random and CVPR are all non-adaptive models that could always get stuck under inexperienced circumstances, with senseless action taken. MPSL abstracts task features only depending on navigation targets, resulting in poor performance when dealing with unseen environments. For BRM and GCN, all the mappings learnt from visual features and semantic features could be quite unstable as the appearance of targets and scenes changing. Results demonstrate that our maml-based learning method could compatibly comprehend complex characteristics of different scenes and targets, bringing about effective adaption to novel tasks.

For ablation study we design and test Our(vis) and Our(oh). These models differ from proposed approach in the data preprocessing phase. For our(vis) we remove semantic features in the model, using visual features abstracted from observation/target images to create mappings. For Our(oh) we try another target encoding method where each word vector loses semantic association with each other. As results shown in Table 1, the success rate of both models generally decrease (27% in Our(vis) and 36% in Our(oh)), which supports our hypothesis that incorporating semantic features and relatedness into the state could be an efficacious way to obtain experience generalized to a larger scale.

Our(loc) can be considered as an scene-exclusive version of Our(glo). According to four types of scenes we train different Our(loc) models, most of which have made better performance than Our(glo). This could be explained by the fact that when training based on all scene types, Our(glo) may reach to an over-fitting situation that it becomes well-trained for one specific scene type, failing to find targets in other types of scenes. The evidence could be found in the data of Scenario 2 (navigating to trained target in unseen environment) in Table 1. Our(glo) obtains its highest success rate as 42.1% in kitchen, while in bedroom/livingroom/bathroom it only attains half accuracy.

It is also worth mentioning that, in the third scenario(navigating to untrained target in unseen environment) agent fails to localize the target object in a relative high success rate as it has achieved in other scenarios. In most of failure cases, the agent may remain static or wander around a specific area, which could not be simply explained by over-fitting or lack of collision information. Through analysis of discrepancy in these experiments, the reason for drop of performance could be attributed to the increasing dimensionality of generalization. Dealing with both unknown tasks and unfamiliar environment brings about more imprecise recognition about mapping relation between targets and its semantic surroundings. Great richness of details among scenes leads to the fact that experience learned from specific task can hardly make sense in others. To solve the problem, we may be devoted into datasets sampling with modest similarity and discuss model's performance in future work.

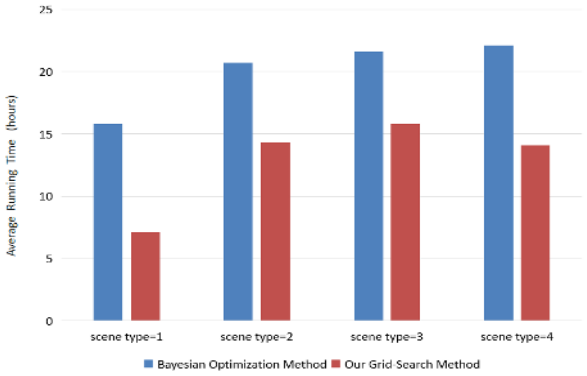

FIGURE 6. Comparison of Bayesian optimization method's time consumption with our Grid-Search method.

D. HYPER-PARAMETER OPTIMIZATION

Notably, as mentioned in the previous section, hyper-parameters appear to be extremely important in our study since they determine how fast model learns and how long model should be trained during meta-training phase. For example the value of hyper-parameter N makes great sense in adjusting model maturity and avoiding over-fitting to the training datasets. In the machine learning field usually hyper-parameters are configured in manual setting based on simple empirical analysis, determined in a computationally efficient manner. Although a few parameter tuning methods have also been introduced into hyper-parameters optimization, such as bayesian optimization [38], SMAC [39] and ParamILS [40], resulting in more precise results, these approaches appear to be inefficient and impractical as the training and testing datasets grow in size, in this case the tedious interacting process proceeds.

Consequently, to achieve higher performance with relatively few computational cost, we design a hyper-parameter tuning method according to Grid Search [41] to obtain appropriate combined value of N, α and β. This method is performed in the bathroom-specific generalization experiment to preliminarily determine hyper-parameters, and then applied in all-scenes experiments. Figure 6 shows the comparison Bayesian optimization method's time consumption with our Grid-Search method. The result demonstrates that the runtime of both hyper-parameter tuning approaches grows gradually as the amount of scene types included in datasets increases, and Grid-Search based method requires less time to obtain available hyper-parameter values. Although Bayesian optimization method proves to be more efficient than Grid-Search, the iteration process restricts it from parallel training. In contrast our Grid-Search based approach could perform multi-thread processing, with a great deal of time consumption reduced. Compared to other hand-crafted or trial-and-error approaches, our grid-search method turns out to be more applicable for navigation tasks.

V. CONCLUSION

In this article, we proposed an vision-language adaptive navigation model to enable agent generalizing to untrained navigation tasks. Our network structure is constructed on the basis of DRL method, which provides appropriate actions according to visual observation and text-defined target. The integration of FCIS network and word vector obtained from Word2vec precisely creates mapping from visual features to semantic features. A novel self-adaptive learning method based on MAML has been proposed to achieve fast adaption to unfamiliar tasks through the meta-adapting phase. To evaluate proposed model's performance, several experiments with three scenarios have been conducted. As results illustrated, our model could accomplish target-driven navigation tasks and generalizing to untrained ones with higher success rate than other models of existing researches. However, the performance still appears to be much worse than the human level, which is probably because that our approach is hardly to be adjusted to accumulate experience in the perfectly efficient way, and also some vital information such as depth has been ignored. In future work we will focus on further decoupling the meta-training phase to improve the generalization ability of current model across tasks and scenes.

REFERENCES

[1] V. Mnih, K. KavuKcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," in *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] 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. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2850–2869.

[3] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 2017.

[4] 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. (ICRA)*, May 2017, pp. 3357–3364.

[5] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. ICML*, vol. 48, 2016, pp. 1995–2003.

[6] 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, Apr. 2020.

[7] A. Taalimi, A. Rahimpour, L. Liu, and H. Qi, "Multi-view task-driven recognition in visual sensor networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2099–2103.

[8] R. K. Vijay and N. T. John, "Actor-critic algorithms," in *Proc. NIPS*. Cambridge, MA, USA: MIT Press, 1999, pp. 1008–1014.

[9] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2359–2367.

[10] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 1–5.

[11] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, Jun. 2005, pp. 539–546.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic metalearning for fast adaptation of deep networks," in *Proc. ICML*, vol. 2, no. 4, 2017.

[13] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, Jun. 2005, pp. 539–546.

[14] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 667–676.

[15] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 572–577.

[16] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[17] D. Kim and R. Nevatia, "Symbolic navigation with a generic map," *Auto. Robots.*, vol. 6, no. 1, pp. 69–88, 1999.

[18] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1403–1410.

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[20] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[21] Y.-H. Kim, J.-I. Jang, and S. Yun, "End-to-end deep learning for autonomous navigation of mobile robot," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–6.

[22] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2616–2625.

[23] D. Misra, J. Langford, and Y. Artzi, "Mapping instructions and visual observations to actions with reinforcement learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1004–1015.

[24] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2054–2063.

[25] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018.

[26] P. Anderson, A. Chang, D. Singh Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," 2018, *arXiv:1807.06757*. [Online]. Available: http://arxiv.org/abs/1807.06757

[27] Y. Liu, Y. Cong, and G. Sun, "Memory-based parameterized skills learning for mapless visual navigation," in *Proc. Int. Conf. Image Process. Image*, Sep. 2019, pp. 1890–1894.

[28] M. Bhardwaj, S. Choudhury, and S. Scherer, "Learning heuristic search via imitation," 2017, *arXiv:1707.03034*. [Online]. Available: http://arxiv.org/abs/1707.03034

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] K. M. Hermann, F. Hill, S. Green, and F. Wang, "Grounded language learning in a simulated 3D world," 2017, *arXiv:1706.06551*. [Online]. Available: https://arxiv.org/abs/1706.06551

[31] Y. Sasaki, S. Matsuo, A. Kanezaki, and H. Takemura, "A3C based motion learning for an autonomous mobile robot in crowds," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1036–1042.

[32] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," in *Proc. RSS*, 2018. [Online]. Available: https://arxiv.org/abs/1802.01557

[33] J. J. Bentley, "Fast algorithms for geometric traveling salesman problems," *INFORMS J. Comput.*, vol. 4, no. 4, pp. 387–411, 1992.

[34] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, "Bayesian relational memory for semantic visual navigation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2769–2779.

[35] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *Proc. Comput. Vis. Pattern Recognit.*, 2018. [Online]. Available: https://arxiv.org/abs/1810.06543

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016, pp. 630–645.

[37] W. C. Anderson, K. Carey, E. M. Sturzinger, and C. J. Lowrance, "Autonomous navigation via a deep Q network with one-hot image encoding," in *Proc. IEEE Int. Symp. Meas. Control Robot. (ISMCR)*, Sep. 2019, pp. A2-2-1–A2-2-6.

[38] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, Dec. 2012, pp. 2951–2959.
[39] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Proc. Int. Conf. Learn. Intell. Optim.*, vol. 2, 2011, pp. 507–523.
[40] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stuetzle, "ParamILS: An automatic algorithm configuration framework," *J. Artif. Intell. Res.*, vol. 36, pp. 267–306, Oct. 2009.
[41] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. Robot. Res.*, vol. 23, nos. 7–8, pp. 673–692, Aug. 2004.

TIANFANG XUE was born in Shenyang, China, in 1993. He received the B.S. degree from Fudan University, China, in 2015. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, China. His research interests include multi-agent systems, artificial intelligence, and machine learning algorithms in visual navigation field.

HAIBIN YU (Senior Member, IEEE) received the Ph.D. degree from Northeastern University, China, in 1997. He has been a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences, China, since 1997, where he is currently serving as the Director. He has published two books, authored or coauthored over 200 articles, and held over 50 patents. His research interests include wireless sensor networks, industrial communication and networked control, industrial automation, and intelligent manufacturing. He and his research team have proposed the WIA-PA and WIA-FA standards which are specified as IEC 62601 and IEC 62948, respectively. He was elected as an ISA Fellow for his contributions in fieldbus technologies, in 2011. He serves as the Chair of *IEC ACART*, the Vice-Chair of the Chinese Association of Automation, and the Chair of the China National Technical Committee for Industrial Process Measurement Control and Automation Standardization.