Visual Perception Generalization for Vision-and-Language Navigation via Meta-Learning

Ting Wang, Zongkai Wu, and Donglin Wang, Member, IEEE

Abstract—Vision-and-language navigation (VLN) is a challenging task that requires an agent to navigate in real-world environments by understanding natural language instructions and visual information received in real time. Prior works have implemented VLN tasks on continuous environments or physical robots, all of which use a fixed-camera configuration due to the limitations of datasets, such as 1.5-m height, 90° horizontal field of view (HFOV), and so on. However, real-life robots with different purposes have multiple camera configurations, and the huge gap in visual information makes it difficult to directly transfer the learned navigation skills between various robots. In this brief, we propose a visual perception generalization strategy based on meta-learning, which enables the agent to fast adapt to a new camera configuration. In the training phase, we first locate the generalization problem to the visual perception module and then compare two meta-learning algorithms for better generalization in seen and unseen environments. One of them uses the model-agnostic meta-learning (MAML) algorithm that requires few-shot adaptation, and the other uses a metric-based meta-learning method with a feature-wise affine transformation (AT) layer. The experimental results on the VLN-CE dataset demonstrate that our strategy successfully adapts the learned navigation skills to new camera configurations, and the two algorithms show their advantages in seen and unseen environments respectively.

Index Terms—Embodied agent, meta-learning, vision-and-language navigation (VLN), visual perception generalization.

I. INTRODUCTION

The vision-and-language navigation (VLN) task requires an agent to follow natural language instructions to navigate in photo-realistic environments according to the visual information captured in real time and the prebuilt navigation graph [1]. In recent years, the VLN task has attracted widespread attention due to its promising real-life applications and many methods have achieved satisfactory results in terms of success rate in simulation environments [2]–[4]. Recently, Krantz et al. [5] break through the limitation of the navigation graph and migrate the VLN task to continuous environments for the first time. Subsequently, combining a subgoal module with the traditional path planning on maps, Anderson et al. [6] transfer the VLN task from the embodied agent in simulation environments to the physical robot in real environments and successfully guarantee an acceptable success rate.

However, due to the limitation of datasets, the above-mentioned agents or robots are trained by a fixed camera configuration, such as 1.5-m height, 90° horizontal field of view (HFOV), and so on. But in fact, robots for different purposes have various forms, and even their camera configurations are more different. As shown in Fig. 1, the camera heights or HFOVs of robots with specific functions are different such as sweeping robots, search and rescue drones, four-legged mechanical dogs, and so on. The huge gap in visual information acquired by cameras with different configurations makes it difficult for learned navigation models to be directly shared among different robots.

To solve the perception generalization problem of heterogeneous robots in VLN, we propose a visual perception generalization strategy based on meta-learning. According to the functions of each module of the VLN model, we divide the overall VLN model into three modules: a visual perception module, a language understanding module, and a navigation reasoning module. Since camera configurations only affect the received image information, we locate the generalization problem to the visual perception module. We hope that the visual perception module can extract as close features as possible in the same location state under different sensor configurations, so that the navigation module can make corresponding consistent action decisions.

To train our visual perception module, we pretrain a VLN navigation model with a classic structure beforehand. Then, the learned visual perception module is used as the supervision information for our meta-learner. We adopt two methods based on meta-learning to train our visual perception module and compare the two methods in seen and unseen environments. For the first method, inspired by a few-shot adaptation of visual navigation [7], we train the visual perception module with the model-agnostic meta-learning (MAML) algorithm, so that the visual perception module can quickly adapt to a new sensor configuration through fine-tuning with a small amount of data. However, it is difficult to obtain even a small amount of data, and low-quality data will bring additional errors. Consequently, we consider another method, that is, adding feature-wise affine transformation (AT) layers to the visual perception module to simulate visual features under different sensor configurations. The AT parameters are trained by learning to learn, so that the visual...
perception module can be better generalized to arbitrary camera configuration without adaptation process.

In the validation phase, for cameras with different heights and HFOVs, we evaluate the above two mentioned methods on the Habitat simulator, involving VLN in continuous environments (VLN-CE) [5] tasks. The experimental results show that our strategy successfully adapts the learned navigation model to new sensor configurations, and the two methods show their advantages in seen and unseen environments respectively. It is worth mentioning that our methods are suitable for visual generalization between sensors with different configurations. In this brief, we only explain our methods from two aspects of camera height and HFOVs.

In general, our main contributions are as following.
1) We take notice of the visual generalization problem brought by different sensor configurations for different robots in VLN tasks and locate the problem to the visual perception module.
2) We propose a visual perception generalization strategy based on meta-learning to deal with the generalization problem. And we compare the two methods in our strategy for better generalization in seen and unseen environments respectively.
3) We implement the visual perception generalization under different camera heights and HFOVs. The experimental results prove the effectiveness of our strategy.

II. RELATED WORK

A. Vision-and-Language Navigation

Anderson et al. [1] first propose the concept of VLN and provide the room-to-room (R2R) dataset collected using the Matterport3-D Simulator based on real images. On this basis, many works have made progress and addressed some of the challenges for the VLN task. Fried et al. [8] propose a speaker model, which generates corresponding natural language instructions according to the sampled new paths for data augmentation and route selection. To tackle the problems of cross-modal grounding and ambiguous feedback, Wang et al. [2] propose a reinforced cross-modal matching (RCM) approach, using a matching critic as an intrinsic reward to encourage global matching between language instructions and routes and using a reasoning navigator with three attention mechanisms to align the global matching between language instructions and routes and using the speaker model [8] to generate natural language and image information inside real indoor environments. Luo et al. [7] and Wortsman et al. [20] combine MAML and visual navigation. The former proposes a self-adaptive visual navigation (SA VN) method that enables the agent to interact with the environment without any additional supervision through meta-reinforcement learning. The latter divides the navigation framework into perception and inference networks and utilizes MAML to train the perception network so that the agent can adapt to new observations with a few shots.

In addition to the above optimization-based content, metric-based learning is also one of the common methods of meta-learning. A metric-based model usually consists of a feature encoder for feature extraction and a metric function for classification tasks. The Matching Net [21] uses Cosine in the embedding space to measure the features extracted from the support set and achieves classification by calculating the matching degree on the test samples. Referring to the clustering idea, prototypical networks [22] project the support set into a metric space to obtain the vector mean and calculate the distance from the test sample to each prototype for classification. The relation module proposed by the relation network [23] replaces Cosine and Euclidean distance metric in the matching net and prototypical networks, making it a learnable nonlinear classifier for judging relations and realizing classification. These are three currently popular metric-based meta-learning models. We draw on the feature-wise AT layers on the feature encoder to simulate the distribution of image features captured by cameras with different configurations to achieve the generalization of visual perception.

III. VISUAL PERCEPTION GENERALIZATION FOR VLN

In this section, we propose a visual perception generalization strategy, which can make the trained VLN agent generalize quickly and effectively to other agents with different sensor configurations.

As described by the framework of the VLN model in Fig. 2, we divide the VLN model into three modules: a visual perception module, a language understanding module, and a navigation reasoning module. We locate the generalization problem to the visual perception module and assume that the action space is consistent for any agent. Hence, we only adapt the visual perception module to new sensor configurations based on meta-learning while freezing the instruction understanding module and the navigation reasoning module.

A. Task Formulation

In the VLN task, given a natural language instruction, the agent’s goal is to navigate toward the target location by co-processing language and image information inside real indoor environments.
navigation reasoning module and language understanding module.
The learned visual perception module is used as the supervision information for retraining with meta-learning, which is to promote the visual perception module to produce as close as possible intermediate visual features under the same location but different observations, that is, \( \hat{\phi}(\hat{o}(s)) \rightarrow \phi(o(s)) \) and \( \hat{\phi}(\hat{o}(s)) \rightarrow \phi(o(s)) \). \( \hat{\phi} \) and \( \phi \) represent the trained feature extractors, and \( \hat{\phi} \) and \( \hat{\phi} \) represent the new feature extractors that can be used with new sensors of different configurations (in the VLN-CE task, the feature extractor usually refers to ResNet50). We can train a new visual perception module by minimizing the loss

\[
\mathcal{L}_r = \sum | \phi(a(s)) - \hat{\phi}(\hat{o}(s)) | \quad (1)
\]

\[
\mathcal{L}_d = \sum | \phi(o(s)) - \hat{\phi}(\hat{o}(s)) | . \quad (2)
\]

Using supervised learning to optimize the visual perception module requires a large number of observations from the target observation space as training data. This makes pure supervised learning time-consuming and infeasible. Therefore, we use meta-learning to train the visual perception module. The role of meta-learning in the visual perception module is shown in Fig. 3. Here, we consider two meta-learning-based methods to train the visual perception module for generalization.

1) Few-Shot Adaptation With MAML: We propose a method with the MAML [19] algorithm, which can quickly adapt navigation agents to new sensors with a very small amount of data and few-shot fine-tuning.

We consider the distribution of VLN tasks \( p(\Gamma) \). Each support set of the task \( \Gamma_i \) consists of \( k \) observation images with specific visual perception functions \( \hat{\phi} \) and \( \phi \). We train our visual perception module with \( \hat{\phi} \) and \( \phi \), which is parameterized by \( \theta \) and \( \mu \), respectively, to learn an unseen test using only \( k \) samples.

The details of the algorithm are shown in Algorithm 1. We randomly sample a batch of tasks \( \Gamma_i (i = 1, \ldots, N) \) from the distribution and calculate adapted parameters \( \theta', \mu' \) using one or more gradient descent updates on task \( \Gamma_i \) in steps 4 and 5. The learning rate \( \alpha \) and \( \beta \) are fixed as hyperparameters. Then, in step 7, the meta-parameters \( \theta \) and \( \mu \) are updated by optimizing for the performance of \( \phi' \) and \( \phi' \) with respect to \( \theta \) and \( \mu \) through the tasks sampled from \( p(\Gamma) \).

2) Generalization With Affine Transformation: MAML-based optimization methods need to go through a few-shot adaptation process when facing new sensors, but obtaining even a little data is still difficult. To this end, referring to metric-based meta-learning, we propose another generalization method that does not require any adaptation for VLN.

We assume \( K \) domains of seen camera configurations \( \{T_1^{\text{seen}}, T_2^{\text{seen}}, \ldots, T_K^{\text{seen}}\} \) available in the training phase. Our goal is that the visual perception module learned from the existing sensor configurations can generalize well to a new sensor configuration. For example, the model we trained using the visual information

Algorithm 1 Few-Shot Adaptation With MAML

Require: Task distribution \( p(\Gamma) \)
Require: Learning rate \( \alpha \), \( \beta \), \( \gamma \), \( \delta \)
1: Randomly initialize \( \theta \) and \( \mu \)
2: while not done do
3: for mini-batch of tasks \( \Gamma_i \in p(\Gamma) \) do
4: Calculate \( \nabla_{\theta} \mathcal{L}_{\Gamma_i}(\theta) \) and \( \nabla_{\mu} \mathcal{L}_{\Gamma_i}(\theta) \)
5: Compute adapted parameters with gradient descent:
\[
\theta' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\Gamma_i}(\theta)
\]
\[
\mu' = \mu - \beta \nabla_{\mu} \mathcal{L}_{\Gamma_i}(\theta)
\]
6: end for
7: Update \( \theta = \theta - \gamma \sum_{i=1}^{N} \nabla_{\theta} \mathcal{L}_{\Gamma_i}(\theta) \) and \( \mu = \mu - \delta \sum_{i=1}^{N} \nabla_{\mu} \mathcal{L}_{\Gamma_i}(\theta) \)
8: end while

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
parameters in the ResNets with the pseudo-seen task and the visual features of the image information obtained by agents with different seen task, we add the AT layers to the two ResNet50 networks to imitate various visual features from Gaussian distributions (4)–(7). In each training iteration, we sample a pseudo-seen domain from a set of seen domains \( \{T^\text{seen}_1, T^\text{seen}_2, \ldots, T^\text{seen}_K\} \) and a pseudo-unseen task \( \Gamma^\text{pu} \) from \( \Gamma^\text{ps} \). To address the above problem, we insert feature-wise AT (AT) \( \phi_{\mu} \) and \( \phi_{\mu, \rho} \) to the two ResNet50 networks to imitate the visual perception module and update the parameters \( \theta \) and \( \mu \) of the ResNets.

In our task, the language understanding and navigation reasoning modules are affected by the change of observation space. Hence, we select two models mentioned in VLN-CE [5] as our baselines, which can understand the natural language instructions and navigate to the target location smoothly based on language and vision input.

We consider two models:

1) Sequence-to-Sequence Model: As shown in the top part of the navigation reasoning module in Fig. 2, the core of the simple sequence-to-sequence (seq2seq) model is a gate recurrent unit (GRU), which takes visual information and language instructions as input and predicts the next action. Through the visual perception module, the semantic visual features \( \phi(o(s_t)) \) (for RGB) and depth information \( \phi(o_d(s_t)) \) (for Depth) can be obtained, respectively, and the last hidden state \( h_t \) of LSTM is applied to encode language instructions as \( \omega = \text{LSTM}(x_1, x_2, \ldots, x_t) \). At time step \( t \), the predicted action \( a_t \) of the VLN agent is expressed as

\[
h_t^{(a)} = \text{GRU}\left(\phi(o(s_t), \phi(o_d(s_t))), \omega, h_{t-1}^{(a)}\right)
\]

We use the way of learning to learn to optimize the hyperparameters \( \theta \) and \( \mu \) of the AT layer, as described in Algorithm 2. In each training iteration, we sample a pseudo-seen \( \Gamma^\text{ps} \) and a pseudo-unseen \( \Gamma^\text{pu} \) domain from a set of seen camera configurations \( \{T^{\text{seen}}, T^{\text{seen}_2}, \ldots, T^{\text{seen}_K}\} \). We then update the parameters in the ResNets with the pseudo-seen task \( \Gamma^\text{ps} \), namely

\[
\theta = \theta - \eta \nabla_{\theta} L^\text{ps}_\theta \left(\phi(o(s_t)), \hat{\phi}_\theta(o(s_t))\right)
\]

where \( \eta \) and \( \zeta \) are learning rates. We measure the generalization performance of the updated visual perception module by two steps: (1) removing the AT layers from the ResNet50 and 2) computing the loss \( L^\text{ps}_\theta \) of visual difference on the pseudo-unseen task \( \Gamma^\text{ps} \), which reflect the effectiveness of the AT layers. Finally, we update \( \theta \) and \( \mu \) by

\[
\theta = \theta - \eta \nabla_{\theta} L^\text{ps}_\theta \left(\phi(o(s_t)), \hat{\phi}_\theta(o(s_t))\right)
\]

where \( \eta \) and \( \zeta \) are learning rates. We measure the generalization performance of the updated visual perception module by two steps: (1) removing the AT layers from the ResNet50 and 2) computing the loss \( L^\text{ps}_\theta \) of visual difference on the pseudo-unseen task \( \Gamma^\text{ps} \), which reflect the effectiveness of the AT layers. Finally, we update \( \theta \) and \( \mu \) by

\[
\theta = \theta - \eta \nabla_{\theta} L^\text{ps}_\theta \left(\phi(o(s_t)), \hat{\phi}_\theta(o(s_t))\right)
\]

where \( \eta \) and \( \zeta \) are learning rates. We measure the generalization performance of the updated visual perception module by two steps: (1) removing the AT layers from the ResNet50 and 2) computing the loss \( L^\text{ps}_\theta \) of visual difference on the pseudo-unseen task \( \Gamma^\text{ps} \), which reflect the effectiveness of the AT layers. Finally, we update \( \theta \) and \( \mu \) by

\[
\theta = \theta - \eta \nabla_{\theta} L^\text{ps}_\theta \left(\phi(o(s_t)), \hat{\phi}_\theta(o(s_t))\right)
\]

where \( \eta \) and \( \zeta \) are learning rates. We measure the generalization performance of the updated visual perception module by two steps: (1) removing the AT layers from the ResNet50 and 2) computing the loss \( L^\text{ps}_\theta \) of visual difference on the pseudo-unseen task \( \Gamma^\text{ps} \), which reflect the effectiveness of the AT layers. Finally, we update \( \theta \) and \( \mu \) by

\[
\theta = \theta - \eta \nabla_{\theta} L^\text{ps}_\theta \left(\phi(o(s_t)), \hat{\phi}_\theta(o(s_t))\right)
\]
provided a precomputed shortest path via low-level actions (move forward 0.25 m, turn left or turn right 15°, and stop). Due to the low-level action space, the average length of the trajectories is 55.88 steps.

The VLN-CE dataset is divided into the training set, the validation set, and the test set. The validation set is further split into two parts: the validation-seen subset, where the paths are sampled from the scenes appeared in the training set, and the validation-unseen subset, where the samples are from unseen environments. For validation-seen environments, the validation subset shares the same scenes with the training set but has new instructions. For validation-unseen and test environments, both instructions and scenes have no overlapping with the training dataset.

B. Metrics

We evaluate the performance for our visual perception generalization strategy using four metrics: the navigation error (NE), the oracle rate (OR), the success rate (SR), and the success rate weighted by the path length (SPL). For specific values, refer to [5].

1) NE measures the average distance (in meter) between the agent’s stopping position in the predicted trajectory and the goal in the reference trajectory.
2) OR is the proportion of the closest point in the predicted trajectory to the target in the reference trajectory within a threshold distance.
3) SR is the proportion of the agent stopping in the predicted route within a threshold distance of the goal in the reference route.
4) SPL is a comprehensive metric method integrating SR and TL that takes both effectiveness and efficiency into account.

C. Implementation Setup

We implement our agent on the Habitat simulator [5]. We download the models with the best results in the open-access website as our baseline parameters. For sequence-to-sequence baseline, DAgger-based [32] training has the best result, for which the nth set of experiments is collected by taking the oracle action with probability 0.75° and the current policy action otherwise. For CMA model baseline, the model with the progress monitor [4], DAgger (with probability 0.75°+1), and data augmentation performs best.

1) Experiments for MAML: For adaptation with MAML, our method implementation uses the same inner learning rate of 2e − 4 and outer learning rate of 2e−4, three shots, ten gradient step updates, and the Adam optimizer. We retrain the visual perception module using agents with a 90° HFOV and three different height (e.g., 0.5, 1.0, and 1.5 m) cameras, or 1.5-m height and three HFOV (e.g., 90°, 120°, and 150°) cameras. The three RGB images and three depth images obtained by the three agents with the same camera walking one step according to a language instruction in three random environments are used as the support set of a task T. We verify our method with adaptation on an agent with 0.2 m, 90° HFOV or 1.5 m, 60° HFOV camera in seen and unseen validation environments, respectively.

2) Experiments for Affine Transformation: For generalization with feature-wise AT layers, our method implementation also uses the learning rate of 2e−4 and the Adam optimizer. All seen visual perceptions T^seen(D = [1, 2, 3]) consist of observations obtained by agents with 0.5, 1.0, and 1.5 m height and 90° HFOV cameras, or 90°, 120°, and 150° HFOV and 1.5-m height cameras. Then, we randomly sample the pseudo-seen T^pseudo and pseudo-unseen T^pu from T^seen in each training episode. We test our method without any adaptation process in seen and unseen validation environments on the agent with 0.2 m, 90° HFOV or 1.5 m, 60° HFOV camera, respectively. The different camera configurations of agents during the training and testing phase are clearly stated in Table I. The visual perception differences of cameras with different configurations are shown in Fig. 5.

D. Comparison of Our Generalization Methods

In order to confirm the existence of generalization problem on different camera configurations, we first test the baselines with the validation dataset. Table II summarizes the results of our baselines in validation unseen environments. We train the seq2seq and CMA baselines on an agent with a 1.5-m height, 90° HFOV camera. Then we test our baselines for agents with a 0.2-m height, 90° HFOV camera (the top part of Table II) and a 1.5-m height, 60° HFOV camera (the bottom half of Table II), respectively. Obviously, whether it is changing the agent’s camera height or HFOV, all of the navigation performance is significantly reduced. Table II shows that there are huge gaps in visual perception under different camera configurations, which makes it hard to effectively generalize the navigation strategy to other agents or robots with different camera configurations.

Table III illustrates the experimental results of using our generalization strategy compared with baselines in the validation seen and unseen environments, where four metrics are used for evaluation. Compared with the baselines, our strategy both achieve a competitive improvement regardless of changing the camera height or HFOV. For changing the camera height from 1.5 to 0.2 m in validation-unseen environments, the MAML-based adaptation method achieves the success rates of 0.14 and 0.23, SPL of 0.12 and 0.20 on the two baselines, seq2seq and CMA, respectively. And the method based on AT layers, respectively, reach SR of 0.14 and 0.24 and SPL of 0.12 and 0.22. The success rates of the two methods we introduce are much higher than the baselines. Although the results of the two methods are very similar to each other, it can be seen that the MAML-based adaptation method performs better in seen environments in most cases. We speculate that the difference of performance in different environments may be caused by few-shot adaptation of MAML. The method with AT always shows relatively...
stable generalization performance because it simulates image features obtained by cameras with different unseen configurations and does not require any adaptation. For changing the camera HFOV from 90° to 60°, the performance of our generalization strategy is similar to the above analysis.

In order to further demonstrate the generalization performance of our proposed strategy, we also try another set of experimental settings opposite to the above settings: from a smaller height of 0.5 m, 1.0 m, 1.5 m to 1.8 m and a smaller HFOV of 60°, 90°, 120° to 180°. Table IV shows that our visual perception generalization strategy is still effective under different experimental settings. We also test the CMA model in the test environments (VLN-CE challenge leaderboard), as shown in Table V. It can be seen that the experimental effect is basically the same as that in the validation-unseen environments.

To sum up, the experimental results show that our strategy has satisfactory performance in generalization of visual perception. In seen environments, the MAML-based method usually performs better, while in unseen environments, the AT-based method and MAML-based method have similar effects.

**TABLE III**

| Change Height:0.2m | Validation-seen | Validation-unseen |
|--------------------|------------------|--------------------|
|                     | NE ↓ | OR ↑ | SR ↑ | SPL ↑ | NE ↓ | OR ↑ | SR ↑ | SPL ↑ |
| Seq2seq baseline   | 10.10 | 0.13 | 0.04 | 0.01 | 9.10 | 0.11 | 0.02 | 0.01 |
| Seq2seq+MAML       | 9.13 (+0.97) | 0.32 (+0.19) | 0.13 (+0.09) | 0.11 (+0.10) | 8.89 (+0.21) | 0.30 (+0.19) | 0.14 (+0.12) | 0.12 (+0.11) |
| Seq2seq+AT         | 9.41 (+0.69) | 0.25 (+0.12) | 0.12 (+0.08) | 0.11 (+0.10) | 8.88 (+0.22) | 0.26 (+0.15) | 0.14 (+0.12) | 0.12 (+0.11) |

**TABLE IV**

| Change HFOV:60° | Validation-seen | Validation-unseen |
|-----------------|------------------|--------------------|
|                 | NE ↓ | OR ↑ | SR ↑ | SPL ↑ | NE ↓ | OR ↑ | SR ↑ | SPL ↑ |
| Seq2seq baseline| 8.59 | 0.17 | 0.13 | 0.12 | 8.82 | 0.20 | 0.10 | 0.08 |
| Seq2seq+MAML    | 9.17 (+0.58) | 0.28 (+0.11) | 0.21 (+0.08) | 0.19 (+0.07) | 9.03 (+0.21) | 0.25 (+0.05) | 0.17 (+0.07) | 0.15 (+0.07) |
| Seq2seq+AT      | 9.03 (+0.44) | 0.26 (+0.09) | 0.19 (+0.06) | 0.17 (+0.05) | 8.59 (+0.23) | 0.25 (+0.05) | 0.18 (+0.08) | 0.16 (+0.08) |

**TABLE V**

| Change Height:1.8m | Validation-seen | Validation-unseen |
|--------------------|------------------|--------------------|
|                     | NE ↓ | OR ↑ | SR ↑ | SPL ↑ | NE ↓ | OR ↑ | SR ↑ | SPL ↑ |
| Seq2seq baseline   | 8.41 | 0.24 | 0.16 | 0.11 | 8.25 | 0.26 | 0.17 | 0.13 |
| Seq2seq+MAML       | 9.10 (+0.69) | 0.27 (+0.03) | 0.23 (+0.07) | 0.18 (+0.07) | 8.63 (+0.38) | 0.28 (+0.02) | 0.20 (+0.03) | 0.19 (+0.06) |
| Seq2seq+AT         | 9.10 (+0.69) | 0.27 (+0.03) | 0.18 (+0.02) | 0.15 (+0.04) | 8.63 (+0.38) | 0.28 (+0.02) | 0.20 (+0.03) | 0.17 (+0.04) |

| Change HFOV:180° | Validation-seen | Validation-unseen |
|------------------|------------------|--------------------|
| Seq2seq baseline | 7.28 | 0.37 | 0.18 | 0.15 | 7.65 | 0.37 | 0.19 | 0.14 |
| Seq2seq+MAML     | 7.64 (+0.36) | 0.40 (+0.03) | 0.29 (+0.11) | 0.27 (+0.12) | 7.54 (+0.11) | 0.38 (+0.01) | 0.28 (+0.09) | 0.27 (+0.13) |
| Seq2seq+AT       | 7.73 (+0.45) | 0.37 (+0.06) | 0.28 (+0.10) | 0.26 (+0.11) | 7.73 (+0.08) | 0.37 (+0.00) | 0.28 (+0.09) | 0.26 (+0.12) |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Our work focuses on the perspective generalization problem caused by different sensor configurations of heterogeneous robots in VLN tasks. To solve this problem, we propose a visual perception generalization strategy based on meta-learning. We divide the VLN model into three modules (visual perception module, language understanding module, and navigation reasoning module) and locates the problem to the visual perception module. Then, we consider two specific methods to train our visual perception module with generalization performance. One is based on the MAML algorithm, which adapts the agent to a new camera configuration with few-shot fine-tuning. The other uses learning-to-learn that trains the visual perception module by different sensor configurations of heterogeneous robots in VLN tasks.

### REFERENCES

1. P. Anderson et al., “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3674–3683.
2. X. Wang et al., “Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6629–6638.
3. H. Tan, L. Yu, and M. Bansal, “Learning to navigate unseen environments: Back translation with environmental dropout,” in Proc. Conf. North, 2019, pp. 2610–2621.
4. C.-Y. Ma et al., “Self-monitoring navigation agent via auxiliary progress estimation,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2019. [Online]. Available: https://arxiv.org/abs/1901.03035
5. J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, “Beyond the Nav-graph: Vision-and-language navigation in continuous environments,” in Proc. Eur. Conf. Comput. Vis. (ECCV). Cham, Switzerland: Springer, 2020, pp. 104–120.
6. P. Anderson et al., “Sim-to-real transfer for vision-and-language navigation,” in Proc. Conf. Robot Learn. (CoRL), 2020, pp. 671–681.
7. Q. Luo, M. Sorkin, and S. Ha, “A few shot adaptation of visual navigation skills to new observations using meta-learning,” 2020, arXiv:2011.03609. [Online]. Available: http://arxiv.org/abs/2011.03609
8. D. Fried et al., “Speaker-follower models for vision-and-language navigation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 3314–3325.
9. X. Wang, W. Xiong, H. Wang, and W. Yang Wang, “Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 37–53.
10. F. Zhu, Y. Zhu, X. Chang, and X. Liang, “Vision-language navigation with self-supervised auxiliary reasoning tasks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10012–10022.
11. W. Zhu et al., “Multimodal text style transfer for outdoor vision-and-language navigation,” in Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics, Main Volume, 2021, pp. 1207–1221.
12. W. Zhang, C. Ma, Q. Wu, and X. Yang, “Language-guided navigation via cross-modal grounding and alternate adversarial learning,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 9, pp. 3469–3481, Sep. 2021.
13. Y. Qin, P. Pan, S. Zhang, A. van den Hengel, and Q. Wu, “Object-and-action aware model for visual language navigation,” in Proc. Eur. Conf. Comput. Vis. Proc. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 920–921.
14. W. Hao, C. Li, X. Li, L. Carin, and J. Gao, “Towards learning a generic agent for vision-and-language navigation via pre-training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 13137–13146.
15. Z. Wu, Z. Liu, T. Wang, and D. Wang, “Improved speaker and navigator for vision-and-language navigation,” IEEE MultiMedia, early access, Feb. 9, 2021, doi: 10.1109/MMUL.2021.3058314.
16. A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, “Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 4392–4412.
17. M. Savva et al., “Habitat: A platform for embodied AI research,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9339–9347.
18. C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Proc. Int. Conf. Mach. Learn. (ICML), 2017, pp. 1126–1135.
19. M. Wortsman, K. Elsani, 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. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6750–6759.
20. O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 3630–3638.
21. J. Snell, K. Swersky, and R. Zintgraf, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4077–4087.
22. F. Sang, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1199–1208.
23. H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang, “Cross-domain few-shot classification via learned feature-wise transformation,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.
24. X. Xu, T. Wang, Y. Yang, L. Zuo, F. Shen, and H. Shen, “Cross-modal attention with semantic consistency for image-text matching,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 12, pp. 5412–5425, Dec. 2020.
25. L. Ye, M. Rochon, Z. Liu, and Y. Wang, “Cross-modal self-attention network for referring image segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 10502–10511.
26. X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu, “Multi-modality cross attention network for image and sentence matching,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10938–10947.
27. L. Peng, Y. Yang, Z. Wang, Z. Huang, and H. T. Shen, “MRA-Net: Improving VQA via multi-modal relation attention network,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Jun. 25, 2020, doi: 10.1109/TPAMI.2020.3004830.
28. P. Lu, H. Li, W. Zhu, J. Wang, and X. Wang, “Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering,” in Proc. AAAI, 2018, pp. 7218–7225.
29. A. Chang et al., “MoveNet3D: Learning from RGB-D data in indoor environments,” in Proc. Int. Conf. 3D Vis. (3DV), Oct. 2017, pp. 667–676.
30. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.
31. S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in Proc. 14th Int. Conf. Artif. Intell. Statist., 2011, pp. 627–635.

### TABLE V

| Change Height | NE↑ | OR↑ | SR↑ | SPL↑ |
|---------------|-----|-----|-----|-----|
| (1.5m → 0.2m) | 8.85 | 0.36 | 0.28 | 0.25 |
| (0.2m)        | 6.47 | 0.16 | 0.05 | 0.04 |
| CMA baseline  | 9.45(2.29) | 0.29(0.13) | 0.19(0.14) | 0.16(0.12) |
| CMA+MAML     | 9.60(3.13) | 0.32(0.16) | 0.20(0.15) | 0.18(0.14) |
| CMA+AT       | 9.89 | 0.20 | 0.10 | 0.08 |

### V. CONCLUSION

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.