

Multimodal Aggregation Approach for Memory Vision-Voice Indoor Navigation with Meta-Learning

†Liqi Yan^{1,3}, †Dongfang Liu², Yaoxian Song^{1,3}, *Changbin Yu³

Abstract—Vision and voice are two vital keys for agents’ interaction and learning. In this paper, we present a novel indoor navigation model called Memory Vision-Voice Indoor Navigation (MVV-IN), which receives voice commands and analyzes multimodal information of visual observation in order to enhance robots’ environment understanding. We make use of single RGB images taken by a first-view monocular camera. We also apply a self-attention mechanism to keep the agent focusing on key areas. Memory is important for the agent to avoid repeating certain tasks unnecessarily and in order for it to adapt adequately to new scenes, therefore, we make use of meta-learning. We have experimented with various functional features extracted from visual observation. Comparative experiments prove that our methods outperform state-of-the-art baselines.

I. INTRODUCTION

Indoor navigation is an indispensable ability of mobile robotic systems to operate on daily tasks. In order to fulfill various missions, it is imperative for the indoor mobile robot to be able to realize highly efficient search, plus location and approach tasks in restricted circumstances [1] [2]. The above capabilities are conducive to a lot of practical applications, enabling higher living standards and enhancing human competence. For instance, such capabilities would enable personal assistant and hazard-extinguishing robots to operate more efficiently under indoor circumstances.

Aiming to make a leap forward in the field of navigation models, many in-depth reinforcement learning (RL) methods have been put forward [3] [4] [5]. Some people describe these as the Partially Observable Markov Decision Process (POMDP) and end-to-end

This work is supported in part by the National Science Foundation China (NSFC) 61761136005, and ARC through DP190100887 and DP160104500.

†These authors contributed equally to this work.

*Corresponding email: yu_lab@westlake.edu.cn

¹L. Yan and Y. Song are enrolled at Fudan University, China

²D. Liu is with Department of Computer Graphics Technology, Purdue University, USA

³L. Yan, Y. Song and C. Yu are with the School of Engineering, Westlake University, China

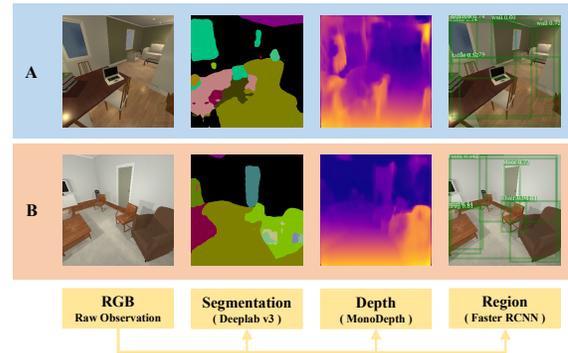


Fig. 1. Two demos of semantic segmentation, depth estimation and object detection from visual observation. Semantic segmentation can identify walkable areas, walls and obstacles, while object detection recognizes targets and depth estimation predicts the distance of all of the above.

policy learning approaches, which are used to connect observation and movements [6] [7] [8] [9] [10]. Yet, for all that, RL algorithms frequently suffer from data hunger for training. A large array of 3D synthetic environments have been created to resolve this issue but the trained models sometimes are unworkable in practical application [11] due to the fact that they struggle to adapt to varied circumstances [12] [13].

There is an inherent chaos in daily life, which deserves our attention. For new daily missions, people are able to find solutions within a relatively short time by applying meta-skills acquired from multiple past activities and experiences. This phenomenon provides an inspiring analogy with the present RL-oriented navigation approaches. Human beings are born with the innate ability to transfer prior knowledge, to learn new skills from experience and to apply them to a variety of missions freely. We aim for machines and robots to be equipped with similar capacities.

Bearing this in mind, we focus on meta-learning [14] [15] and transfer learning [16] [17]. The former can realize efficient learning with a reduced amounts

of data, while the latter could facilitate the learning of new tasks via the application of knowledge already acquired from relevant familiar activities. In contrast, those other representative meta-learning approaches [14] [15] require large amounts of manually designed missions, which is unfeasible for embodied agents.

In this paper, we introduce a novel RL unsupervised method, capable of generating various missions. As mentioned, we call the proposed method, the Memory Vision-Voice Indoor Navigation (MVV-IN). In our design, the agent could acquire transferable meta-skills proficiently so, as to cope with new situations by applying meta-skills to navigate them. As Fig. 1 shows, our model uses multi-modal aggregation, which integrates voice target information into the network and also integrates spatial and semantic information obtained by monocular vision. In addition, we apply advanced attention mechanisms to the agent’s perception of the object region, so that the same can focus on important objects. The model includes memory able to remember short and even long-term states, so, as to avoid the loss of important relevant information. We perform our experiments in various settings and experimental results show that the method significantly outperforms the baseline in all of the metrics and in inference speed.

In summary, contributions of our work include: (a) a proposal of a new navigation task combining vision and voice, (b) the feeding and aggregation of multimodal visual information into the navigation network, (c) the proposal of region self-attention in visual environment understanding and navigation, (d) robust experimental proof that multimodal visual information improves significantly the navigation performance.

II. RELATED WORK

The advancement of computer vision has opened the door to considerable effort directed to use deep-learning approaches for indoor navigation. [4] has created a type of indoor navigation, in which the mobile robot is provided with the image of its object; [13] has solved difficulties in indoor navigation via the training of a dual mapper and planner model; [12] accelerates RL training of indoor navigation under the auspices of loop closure detection; [18] added a topological map to his study of indoor navigation missions; [19] devised a deep RL model that could realize autonomous supervision for navigation; [20] anticipated navigation strategies by means of target inspectors and semantic segmentation models; and [21] [22] have integrated the semantic information to improve the generalization for

unseen settings. All the above perspectives of research have provided reference to our studies.

Our approach differs from others in the specific updated parameters used during navigation and the ways to achieve it. Applications as autonomous drive [23], map-oriented city navigation [24] and gameplay [25] have been probed into learning-centered navigation. In addition, language-training-based navigation has been studied [26] [27] [28]. In our paper, we aim at applying meta-learning to navigate new scenes more successfully.

III. OUR APPROACH

A. Task Definition

Vision-Voice Navigation task. Given a command by voice V , e.g. “Please move to the television”, our goal is to navigate in a room. The agent should start from a random location l_0 and use only RGB pictures taken by monocular camera from the first perspective. Inspired from Seq2Seq (Sequence-to-Sequence) [29] model, we formulate the vision-voice task $t_{mrv-in} \in T_{mrv-in}$ as a tuple $t_{mrv-in} = (V, l_0, r_0)$, where r_0 is the random starting orientation. Generally, the MVV-IN system receives the voice signal V as well as a sequence of images $\mathcal{I} = (I_0, \dots, I_q)$ when the agent interacts with the environment, and carries out a sequence of actions as policy $\mathcal{P} = (a_0, \dots, a_q)$ after $q+1$ steps, where $a \in A$ and A is a set of actions the agent can perform. The system estimates the probability:

$$P(\mathcal{P}|V, l_0, r_0) = \prod_{i=0}^q P(a_i|V, a_{<i}, I_{<i}) \quad (1)$$

where $I_{<i}$ and $a_{<i}$ represents all the visual observations and issued actions before the i^{th} step. The issued proposal \mathcal{P} with the highest estimate should be able to reach the target object with the minimum steps.

Meta-learning task. Meta-learning is to learn how to train a model. We need to train and test the model for visual-voice navigation separately under various scenes (rooms). A complete training and testing process is called a meta learning task. Formally, suppose there are q scenes in total, we define a set of scenes $S = \{s_0, \dots, s_q\}$, so the meta-learning task $\tau_{ml} \in T_{ml}$ can be denoted by the tuple $\tau_{ml} = (s, t_{mrv-in})$, where $s \in S$ and t_{mrv-in} is the MVV-IN task we defined earlier. We divide S into two disjoint sets S_{tr} and S_{te} , for the training tasks T_{tr} and T_{te} respectively. Our task is to let the model find the initialization parameter values that are good for training and testing in all different scenes.

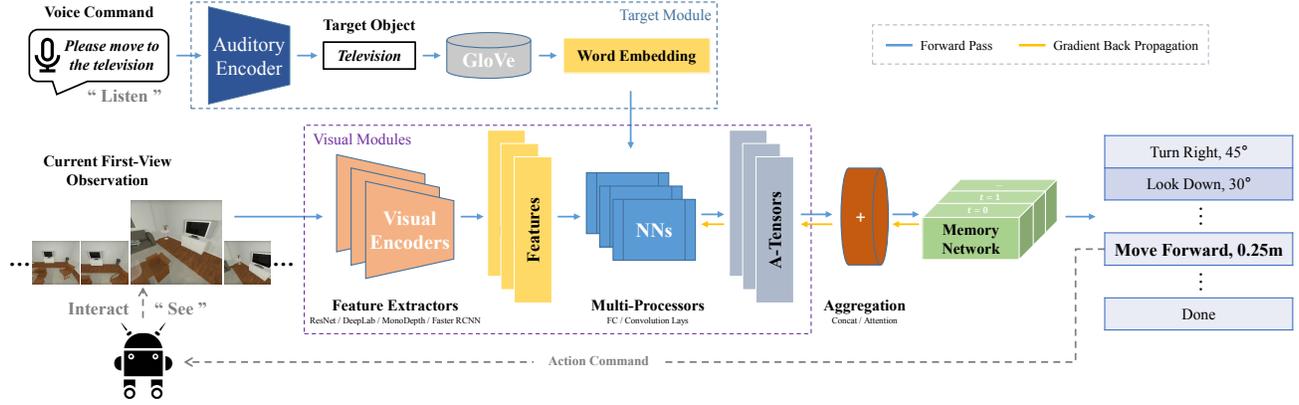


Fig. 2. The pipeline of our proposed MVV-IN system. The current visual observation is input to different encoders to extract different features, like RGB feature and depth feature. After multi-processors, aggregatable tensors (A-Tensor) converted from those features and word embedding, are aggregated and inputted to the memory network. According to its internal state, the memory cell outputs an action policy and its value, like "Move Forward by 0.25m", which will be executed by the agent.

B. Model Design

Our model is modified from the baseline SAVN [3] architecture, which can be described with encoder-decoder [30] architecture, as shown in Fig. 2. Encoder can be divided into two types: auditory encoder and visual encoder. Both of their output needs to be converted into A-tensor through the processors. Compared with the original network structure, the main contribution of our paper is the design of multi-processor, which can realize multi-modal fusion and attention mechanism.

The auditory encoder is to recognize the object target from the speech voice. It consists of two parts, one is language recognition module, the other is target extraction module. The first is responsible for translating the language into the text, while the second is responsible for extracting the target $o \in O$ from the text according to the given rules.

The visual encoder is to mine the deep information of visual observation and extract specific feature from it. We use several different visual analysis decoders to extract different visual features. These features include general RGB feature $F^{(RGB)}$, semantic segmentation feature $F^{(s)}$, depth feature $F^{(d)}$, and object region feature $F^{(o)}$. At the meantime, object region-box proposals $B^{(o)}$ are also predicted and aggregated in our model.

As Fig. 2 shows, our visual module contain the visual encoder as well as the multi-processor, which is to convert the feature of image into an aggregatable tensor (A-Tensor). Those A-Tensors is to be mix together and condensed into an embedding using multimodal aggregation. Formally, the embedding obtained by the

multimodal aggregation method can be denoted as:

$$e_i = \gamma(F_i^{(RGB)}, F_i^{(s)}, F_i^{(d)}, F_i^{(o)}, B^{(o)}) \quad (2)$$

The decoder takes the responsibility of memory. So we utilize memory network as decoder, whose mission is to choose the action with the maximal probability at step i under the condition (h_i, e_i, a_{i-1}) and the network parameters, where h_i is the stored hidden state.

C. Multimodal Aggregation

Vision and hearing are two different modal perceptions. At the same time, information from a variety of different modalities can be analyzed from the visual observation. So how to deal with multi-modal data and aggregate them together is a key consideration for our model.

Before aggregation, we need to convert these different features into A-Tensors, as shown in Fig. 3. The details of the various visual feature encoders will be introduced in the experimental part.

Since the value distribution and shape of each feature are different, we adopt different modules as the multi-processors to handle different forms of weighted feature maps. Suppose we need to deal with a feature map $F = [f_{i_0, i_1, \dots, i_t}]_{d_0 \times d_1 \times \dots \times d_t} \in \mathbb{R}^{d_0 \times d_1 \times \dots \times d_t}$ whose shape is a t -dimensional vector $(d_0, d_1, \dots, d_t) \in \mathbb{N}^t$.

In the case of $t = 0$, the feature map is a d_0 -dimensional vector, then the linear layer is more suitable for the conversion module. Then the k -channel aggregatable tensor converted from the vector is given by:

$$T = \text{ReLU}(F \cdot W_{Linear} + B_{Linear}) \quad (3)$$

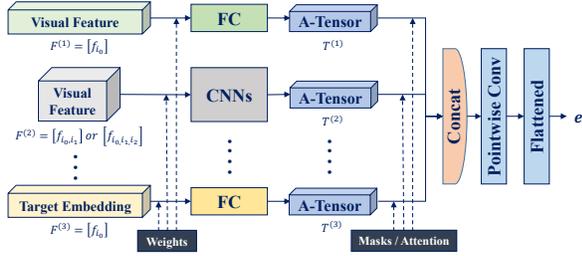


Fig. 3. Multi-processors and multimodal aggregation.

where $W_{Linear} \in \mathbb{R}^{d_0 \times k}$, $B \in \mathbb{R}^k$ and the rectified linear unit (ReLU) is adopted as the activation function.

When $t = 2$ or $t = 3$, and if this feature map F has planar texture characteristics, shape characteristics and spatial relationship characteristics, we can use the convolution layers to further extract those characteristics which is helpful to the navigation system. The case of $t = 2$ can be regard as a special situation of $t = 3$ where $d_0 = 1$. We consider the d_0 dimension as the channel, then A-Tensor is obtained through one or more convolutional layers on the remaining two dimensions:

$$T = \text{ReLU}(G_{Conv}^{(\mu)}(F)) \quad (4)$$

where $G_{Conv}^{(\mu)}$ denotes μ -layer convolutional structure. In the initialization of these convolution layers, the weight is multiplied by a scaling factor due to the use of ReLU, which is recommended as $\sqrt{2}$.

Notation. Denote the set of natural numbers, n -dimensional real vector space and $n \times n$ dimensional real matrix space by \mathbb{N} , \mathbb{R}^n and $\mathbb{R}^{n \times n}$.

To aggregate all those A-Tensors, we concatenate them and input them into a pointwise convolution layer, as shown in the Fig. 3. We reshape the output of this convolution layer to a vector to obtain the final embedding e .

D. Region Attention Module

When people are walking or driving, they often see only local areas in the picture, such as obstacles, target objects, etc., which will affect the next action. We use the self-attention mechanism to select those areas in the observation which the agent really cares about during the navigation steps. The region attention dynamically attends on the region features to form a high-level impression of the visual content and feeds it to the memory network for decoding.

The self-attention mechanism allows the inputs to interact with each other and find out who they should pay more attention to. In our region attention module, we use a double-layer linear layer to obtain the

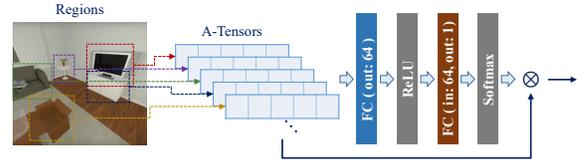


Fig. 4. The self-attention module for region features.

attention scores. As shown in Fig. 4, the softmaxed attention scores for each input is multiplied with its corresponding value.

E. Learning Objective

The MVV-IN system optimize network parameters with two objective loss functions, one is called interaction loss \mathcal{L}_{int} and the another is navigation loss \mathcal{L}_{nav} .

Our meta-learning approach is based on MAML [15] algorithm, which optimize the network parameters on a huge set of tasks \mathcal{T}_{tr} . Each task $\tau_{tr} \in \mathcal{T}_{tr}$ has a tiny meta-training dataset and meta-validation dataset. In our navigation task, agent is designed to be continually learning as it interacts with the environment. So the meta-training dataset is defined as the observed images, hidden states and the issued actions for the first k steps of the agent's trajectory, denoted by $\mathcal{D}_{\tau_{tr}}^{int} = (I_{<k}, h_{<k}, a_{<k})$ according the section A, while the the meta-validation dataset is $\mathcal{D}_{\tau_{tr}}^{nav}$, which represents the counterparts for the remainder of the trajectory.

During this interaction process, our algorithm modifies the navigation policy to minimize the interaction loss \mathcal{L}_{int} , which is a self-supervised loss. In order to assist the agent to learn this objective, we structure a convectional neural network parameterized by ϕ to compute it, which is denoted by \mathcal{L}_{int}^{ϕ} . This network is trained to imitate the \mathcal{L}_{nav} , which is navigation loss defined with a reward for finding the object and a punishment for taking a step. Then the training objective for our MVV-IN system is given by:

$$\arg \max_{\theta, \phi} \sum_{\tau_{tr} \in \mathcal{T}_{tr}} \mathcal{L}_{nav}(\theta - \psi \nabla_{\theta} \mathcal{L}_{int}^{\phi}(\theta, \mathcal{D}_{\tau_{tr}}^{int}), \mathcal{D}_{\tau_{tr}}^{nav}) \quad (5)$$

where the network parameters θ is optimized and updated to $\theta - \psi \nabla_{\theta} \mathcal{L}_{int}^{\phi}(\theta, \mathcal{D}_{\tau_{tr}}^{int})$ after k steps with respect to the self-supervised loss.

IV. EXPERIMENTS AND RESULTS

We articulate our experiments details and results in the following sections.

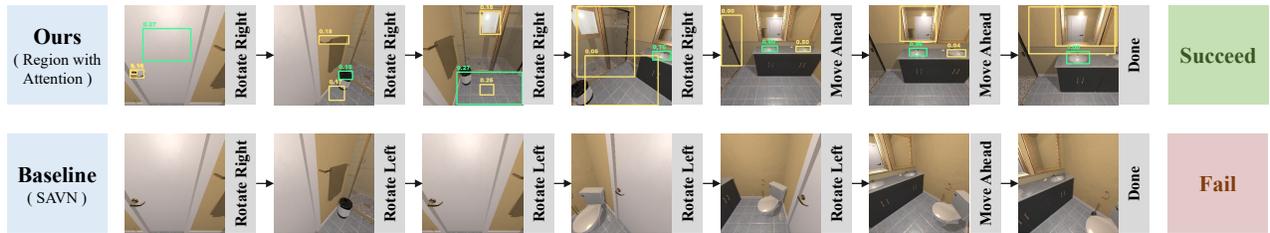


Fig. 5. Comparison with baseline in a sample navigation task: "Move to the sink".

A. Experiment Setup

Dataset. We evaluated the performance of our MVVIN system and its variants on the SAVN dataset using the AI2-THOR environment [31], which includes 4 different room types. Each room type provides 30 room instances. We use 20 rooms for training, 5 rooms for validation and testing respectively in each room type. We perform inference for 1000 difference tasks (50 for each room).

Auditory encoder. To transfer voice to text, we use a novel open source speech recognition model called wav2letter++ [32]. Next, to obtain the target from the command, we use the NLTK package (Natural Language Toolkit) [33] to recognize the noun word in the text.

Memory network. Following the multimodal aggregation, the output is then flattened from the shape of $(64, 7, 7)$ to a 64-dimensional vector, and given as input to a Long Short-Term Memory (LSTM) cell [34]. The hidden state size of LSTM is set to 512.

Target objects. We use the same target objects as SAVN [3]. In different room types, the target object set O is different. For example, in a kitchen, there are cooking utensils, such as a toaster and microwave, while in a bathroom, there are washing and body care supplies, like toilet paper and a liquid soap bottle. We embed the word of the target object with GloVe [35].

Actions. The action set A includes three movement-rotation commands, two camera commands and one "Done" signal. The movement-rotation commands are "Move Ahead", "Rotate Left" and "Rotate Right", and the camera commands are "Look Down" and "Look Up". The horizontal rotation occurs in steps of 45 degrees while the camera commands change the first-person camera incline angle by 30 degrees. Whether the agent completes a navigation task successfully or not, depends on whether, when the "Done" signal is raised, the target object is within 1 meter of the agent and within its field of view.

Training objective. The navigation loss function \mathcal{L}_{nav} is defined with a reward of 5 for finding the object and a punishment -0.01 for taking a step. During the back propagation, Stochastic Gradient Descent (SGD) is used for the interaction gradient updates while Adam is adopted for navigation gradients.

B. Multimodal Feature Extraction and Conversion

Four different visual encoders are used to generate the corresponding visual features from the robot camera observation, as follows:

RGB feature. We use the ResNet-18 [36] pretrained on ImageNet to extract a general image feature from the conv5_4 layer. The shape of this feature is $(512, 7, 7)$.

Segmentation feature. We employ Deeplab [37] to extract segmentation features. Specifically, the feature dimension of the last convolution layer is $(2048, 38, 38)$. We average its second and third dimensions to get a 2048 dimensional vector.

Depth feature. MonoDepth [38] is adopted for depth features. Trained on Cityscapes [39] and KITTI [40], the depth feature map is extracted with a shape of $(384, 512)$.

Region proposal and feature. Faster-RCNN detector [41] with ResNeXt-101 backbone [42] is used for region proposal and feature extraction [23]. The detector is pretrained on Visual Genome [43], and the region features are extracted from the fc6 layer. We select the 7 object regions with the highest confidence value, which are not background, each region feature is a 2048 dimensional vector. At the same time, we also consider to input the region proposal (position and size) into the network, which can be recorded as a 4-dimensional vector.

To fuse the aforementioned features into A-Tensors, we applied different processors for each feature, whose architectures are illustrated in the Table I. RGB, depth and segmentation features all use convolution layers in their conversion, while others use linear layers. For

Feature Type	Dim (In)	Layer(s)	Dim (Out)
<i>RGB</i>	512	1 × 1 conv, 1 × 1 stride	64
<i>Segmentation</i>	2048	FC	64
<i>Depth</i>	1	3 × 3 conv, 3 × 3 stride; 3 × 3 conv, 2 × 3 stride; 3 × 3 conv, 2 × 3 stride; 3 × 3 conv, 2 × 1 stride; 3 × 3 conv, 2 × 2 stride	64
<i>Region (Feature)</i>	2048	FC; Self-Attention	64
<i>Region (Proposal)</i>	4	FC	10
<i>Target</i>	300	FC	64
<i>Action</i>	6	FC	10

TABLE I. The architecture settings of multi-processors in our experiments.

the A-Tensors of 7 regions, we use the self-attention mechanism to calculate the attention scores.

C. Evaluation Metrics

To evaluate our model, we use two metrics that illustrate the superiority of our method: 1. success rate; and 2. success weighted by path length (SPL). Suppose that N is the number of episodes, S_i , L_i and P_i denotes a binary indicator of success, the path length and the optimal trajectory length at episode i in that scene. Then, the success rate is defined as $\frac{1}{N} \sum_{i=1}^N S_i$, while the SPL is defined as $\frac{1}{N} \sum_{i=1}^N S_i \frac{L_i}{\max(L_i, P_i)}$.

D. Quantitative Results

To compare the performance, we perform inference for 1000 different episodes (250 for each room type) for evaluation. Table II illustrates the validation results of our various approaches versus the baseline method's results (SAVN). In all of these experiments, the specific feature is aggregated with the feature of general RGB as well as target word embedding. We provide results for all targets 'All' and a subset of targets whose optimal trajectory length L is greater than 5.

From these metrics results, we can see that all of our methods outperform the baseline, which only uses RGB feature. Experiments show that using self-attention in region feature aggregation can improve the success rate and SPL for 'All'. At the same time, the highest SPL for ' $L \geq 5$ ' can be obtained by feeding the region proposal of object recognition into the network. In general, the best performance is provided by aggregating the features of RGB, segmentation, and region together.

E. Qualitative Demonstrations

A sample navigation task is shown in Fig. 5. We have marked the top three regions of the self-attention

	All		$L \geq 5$	
	SPL	Success	SPL	Success
Baseline - SAVN	13.86	40.20	12.27	25.25
Segmentation	15.52	45.50	12.46	27.24
Depth	14.85	43.40	12.69	26.41
Region (Feature)	15.26	46.00	12.03	28.07
Region (Feature) with Self-Attention	16.04	46.50	12.02	27.74
Region (Proposal)	15.55	46.10	<u>13.83</u>	<u>29.57</u>
Segmentation + Region (Proposal)	<u>17.04</u>	<u>47.00</u>	14.86	30.90
Segmentation + Region (Feature)	17.27	48.70	13.63	30.90

TABLE II. Quantitative results (%).

score. The light green box is the one where the agent's attention is most focused on, as it has the highest self-attention score. It can be seen that without attention (using the baseline method), the agent loses direction and cannot recognize the target "sink" using the baseline method, while our method can quickly help the agent discover and locate the target and judge whether there are obstacles in front.

Note that except for the region corresponding to the target, the attention scores of other regions are fairly low. In addition, the scores of the two "sink" regions are both 0.5 when they first appear. As the agent approaches step by step, the score of the region corresponding to the nearest sink is higher. This illustrates how, by using region attention module, our MVV-IN system can effectively find the closest target object. More details are shown in the accompanying video: https://youtu.be/sO_TIKu0TPk.

F. Time Efficiency and Abolition Study

As shown in Fig. 6, aggregating the depth feature into the network, the region features together with the corresponding proposals, or the segmentation feature together with region proposals (or features with self-attention), all can reduce the inference time during the test. All the above experiments were performed on NVIDIA 2080 Ti GPUs.

Comparing the total time in Fig. 6, we find that it is faster for inference to introduce the feature and proposal of the region into our network than if any of them would be introduced separately. Similarly, Table II shows that adding both segmentation and region features achieves a better performance than adding one of them separately. All our models of variations outperform the baseline, which uses only RGB features. This illustrates the effectiveness of multimodal aggre-

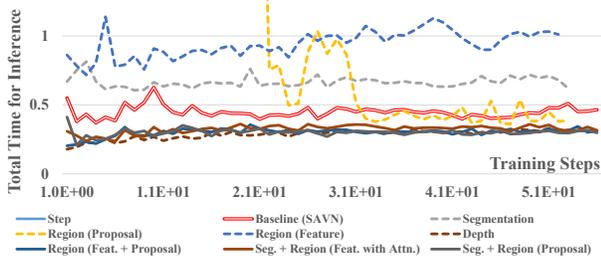


Fig. 6. Total inference time curve for evaluation. The solid line indicates that two features are aggregated into the network together.

gation, including various visual information for indoor navigation.

V. CONCLUSION

In this study, we present a novel Memory Vision-Voice Indoor Navigation (MVV-IN) system. We use multimodal visual info to facilitate the environment’s understanding in navigation systems, using only a monocular camera. Meta-learning technology is used during training so that our model is more robust to unseen scenes and novel tasks. We use extensive experiments to evaluate our method and demonstrate that it is competitive to the existing best approaches. Experimental results show that the proposed method achieves a higher success rate, SPL, and inference speed than the state-of-the-art baseline. In the future, we will evaluate the learning performance of our approach in real-world environments using a programmable mobile robot.

REFERENCES

- [1] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, “On evaluation of embodied navigation agents,” *arXiv preprint arXiv:1807.06757*, 2018.
- [2] D. Liu, Y. Wang, K. E. Ho, Z. Chu, and E. Matson, “Virtual world bridges the real challenge: Automated data generation for autonomous driving,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 159–164.
- [3] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, “Learning to learn how to learn: Self-adaptive visual navigation using meta-learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.
- [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 *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.
- [5] J. Li, S. Tang, F. Wu, and Y. Zhuang, “Walking with mind: Mental imagery enhanced embodied qa,” in *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, 2019, pp. 1211–1219.

- [6] A. Mousavian, A. Toshev, M. Fišer, J. Koščeká, A. Wahid, and J. Davidson, “Visual representations for semantic target driven navigation,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.
- [7] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–53.
- [8] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, “Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.
- [9] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, “Minos: Multimodal indoor simulator for navigation in complex environments,” *arXiv preprint arXiv:1712.03931*, 2017.
- [10] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, “Building generalizable agents with a realistic and rich 3d environment,” *arXiv preprint arXiv:1801.02209*, 2018.
- [11] H. Yu, X. Lian, H. Zhang, and W. Xu, “Guided feature transformation (gft): A neural language grounding module for embodied agents,” *arXiv preprint arXiv:1805.08329*, 2018.
- [12] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, “Cognitive mapping and planning for visual navigation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.
- [13] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, “Learning to navigate in complex environments,” *arXiv preprint arXiv:1611.03673*, 2016.
- [14] A. Nichol, J. Achiam, and J. Schulman, “On first-order meta-learning algorithms,” *arXiv preprint arXiv:1803.02999*, 2018.
- [15] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.
- [16] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 2010, pp. 242–264.
- [17] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.
- [18] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-parametric topological memory for navigation,” *arXiv preprint arXiv:1803.00653*, 2018.
- [19] G. Kahn, A. Villafior, B. Ding, P. Abbeel, and S. Levine, “Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [20] A. Toshev, A. Mousavian, J. Davidson, J. Kosecka, and M. Fiser, “Visual representations for semantic target driven navigation,” 2018.
- [21] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, “Learning and planning with a semantic model,” *arXiv preprint arXiv:1809.10842*, 2018.
- [22] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, “Visual semantic navigation using scene priors,” *arXiv preprint arXiv:1810.06543*, 2018.

- [23] D. Liu, Y. Cui, Y. Chen, J. Zhang, and B. Fan, "Video object detection for autonomous driving: Motion-aid feature calibration," *Neurocomputing*, 2020.
- [24] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2042–2048.
- [25] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," in *Advances in neural information processing systems*, 2017, pp. 5279–5288.
- [26] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo vision," in *Experimental Robotics*. Springer, 2008, pp. 179–190.
- [27] H. Yu, H. Zhang, and W. Xu, "Interactive grounded language acquisition and generalization in a 2d world," *arXiv preprint arXiv:1802.01433*, 2018.
- [28] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.
- [29] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, 09 2014.
- [30] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 09 2014.
- [31] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.
- [32] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "wav2letter++: The fastest open-source speech recognition system," *arXiv preprint arXiv:1812.07625*, 2018.
- [33] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 01 2009.
- [34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [35] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [38] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," *arXiv preprint arXiv:1806.01260*, 2018.
- [39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.
- [40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [42] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
- [43] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.