This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Self-Motivated Communication Agent for Real-World Vision-Dialog Navigation

Yi Zhu[2]*, Yue Weng[1]*, Fengda Zhu[3], Xiaodan Liang[1]†, Qixiang Ye[4], Yutong Lu[1], Jianbin Jiao[4]
[1]Sun Yat-sen University [2]Noah's Ark Lab, Huawei Technologies
[3]Monash University [4] University of Chinese Academy of Sciences

Abstract

Vision-Dialog Navigation (VDN) requires an agent to ask questions and navigate following the human responses to find target objects. Conventional approaches are only allowed to ask questions at predefined locations, which are built upon expensive dialogue annotations, and inconvenience the real-word human-robot communication and cooperation. In this paper, we propose a Self-Motivated Communication Agent (SCoA) that learns whether and what to communicate with human adaptively to acquire instructive information for realizing dialogue annotation-free navigation and enhancing the transferability in real-world unseen environment. Specifically, we introduce a whether-to-ask (WeTA) policy, together with uncertainty of which action to choose, to indicate whether the agent should ask a question. Then, a what-to-ask (WaTA) policy is proposed, in which, along with the oracle's answers, the agent learns to score question candidates so as to pick up the most informative one for navigation, and meanwhile mimic oracle's answering. Thus, the agent can navigate in a self-Q&A manner even in real-world environment where the human assistance is often unavailable. Through joint optimization of communication and navigation in a unified imitation learning and reinforcement learning framework, SCoA asks a question if necessary and obtains a hint for guiding the agent to move towards the target with less communication cost. Experiments on seen and unseen environments demonstrate that SCoA shows not only superior performance over existing baselines without dialog annotations, but also competing results compared with rich dialog annotations based counterparts.

1. Introduction

The richness and generalizability of natural language have significantly boosted the prosperity of navigation tasks in which the agents are encouraged to navigate in the indoor environment to reach the target [1, 27, 3, 34]. In particular, the Vision-Dialog Navigation (VDN) [26, 35, 20, 18], where the question-answering dialogs can be conducted in a human-robot communication manner to facilitate the navigation, has attracted increasing attention in the field of visual navigation.

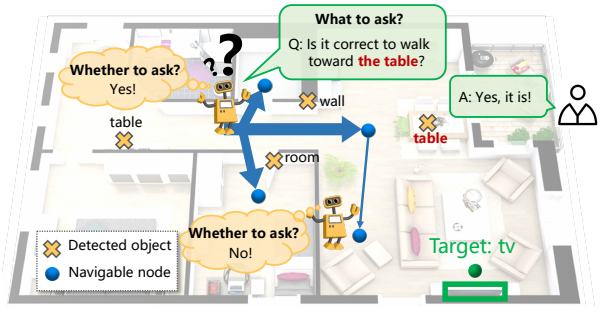

Figure 1: Our Self-Motivated Communication Agent (SCoA) learns whether and what to communicate with human adaptively to acquire instructive information for guiding the navigation without using any dialog annotations.

Cutting-edge practice implements VDN on the premise of tremendous dialog annotations and the dialog occurs in hand-crafted locations during navigation. For example, [26, 35] require to annotate question-answering pairs at fixed locations of the environment in advance to assist the agent training. On the contrary, Roman *et al*. [20] utilized these dialog annotations to pre-train a language model, which is then plugged into the navigation model for dialog generation at every given step interval. In [18], dialog happens only if the agent enters a pre-annotated assistant zone where strong language instructions and image views are provided by the oracle to guide the agent to move towards the target.

Despite the progress, the massive dialog annotations required in existing researches result in two major drawbacks that barricade the real-world deployment of the trained agents: First, the communication is inflexible since the agents are only allowed to ask questions at predefined locations that may contain bias induced by annotators, not at the time when communication is needed. Second, the learning cost is expensive since existing methods are built upon large amounts of labor-intensive dialog annotations. To solve the above problems, we argue that the agent should be able to adaptively communicate with the oracle if necessary,

*Equal contribution.
†Corresponding author.

and such communication should be built upon no or fewer human-annotated dialogs to fit real-world applications.

To this end, we propose Self-Motivated Communication Agent (SCoA), which, as illustrated in Fig. 1, learns to decide whether and what to communicate with human to acquire instructive feedback for guiding the navigation when unsure where to go. As shown in Fig. 2, our SCoA comprises two major components including a whether-to-ask (WeTA) module which learns to predict whether to communicate with the oracle, and a what-to-ask (WaTA) module which learns to generate an informative question for moving towards the target. Specifically, WeTA aims to learn a probability which is used to suggest the agent communicating with the oracle when it is uncertain of which action to take. We propose to model the uncertainty by computing the entropy of the action probability distribution, which in turn serves as the pseudo label to guide the learning of our WeTA. As for WaTA, it first generates some question candidates on-the-fly based on a small set of direction-related sentences as references[1] to get rid of the expensive dialog annotations. By considering the features of language information of the target and vision information of the views, we build a question score vector to pick up the most beneficial question for navigation. By considering the features of question candidates and the optimal next-step view observed by the oracle, an answer score vector is further introduced, which plays as a teacher to guide the learning of the question scores. In this way, our agent can navigate in a self-Q&A manner even when the human is invisible in real-world unseen environment.

Beyond the uncertainty constraint on whether to ask at each navigation step, we formulate the communication as well as the navigation in a unified imitation learning and reinforcement learning framework, which is further equipped with a communication frequency penalty and a navigation progress reward. As result, the agent can reach the target with as little communication cost as possible.

The main contributions in this paper are three-fold:

- Our SCoA tackles the challenging problem of inflexible and annotation-dependent communication for real-world vision-dialog navigation by learning to adaptively determine whether and what to communicate with human to acquire instructive feedback for navigation.
- The communication and navigation are jointly optimized in the unified framework comprising imitation learning and reinforcement learning, to drive the agent to reach target position with less communication cost.
- The performance of our SCoA is demonstrated to be superior over the baselines without using dialog annotations, and even comparable to the counterparts with rich dialog annotations, which proves the ability of our SCoA to generate informative questions for navigation.

[1]The size of our sentence set is around twenty, a magnitude reduction compared with the ten thousand dialog annotations in existing researches.

2. Related Work

Vision-Language Navigation. Different from VDN task which implements navigation in a robot-human communication manner, the vision-language navigation (VLN) [1, 27, 3, 25, 33, 34, 12] requires the agent to interpret a once-for-all natural-language instruction to reach the target. To foster the community development, Anderson *et al.* [1] introduced the first VLN benchmark considering both photo-realistic environment and human natural language. Since then, various methods have been proposed. RCM [27] enforces cross-modal grounding and the self-supervised imitation learning is combined to enhance the generalizability. To overcome the limited seen environment, Fried *et al.* [3] introduced a speaker model and a panoramic representation to augment the data, while EnvDrop [25] produces "unseen" triplets of environment, path and instruction to mimic unseen environment. New paths and instructions are generated in a self-supervised manner. Hao *et al.* [5] complemented a pre-trained model on a large amount of image-text-action triplets for generic representations of visual environment and language instructions.

Object Navigation. Object navigation requires an agent to explore the room and find the target object accurately and efficiently without cooperation with human users [28], which differentiates from robot-human VDN tasks as well. Shen *et al.* [23] devised a fusion scheme for realizing diverse visual representations including RGB features, depth features, segmentation features and so on. To learn viewpoint and target invariant visual servoing for local mobile robot navigation, Li *et al.* [11] trained the Q-learning based network in an end-to-end manner, which also helps improve the robustness of the performance. A hierarchical two-layer structure was proposed by Ye *et al.* [31] in which the high-level layer plans over the sub-goal, and the low-level layer plans over the atomic actions to achieve the goal position. In [14], the 3D knowledge graph and sub-targets are integrated into a unified reinforcement learning framework.

Learning by Asking Questions. Recent advances beyond the navigation also learn to accomplish their tasks by asking questions to the oracle [16, 10, 21, 30, 20, 2, 22]. For example, Vries *et al.* [2] introduced GuessWhat?! game to locate an unknown object in the given image by asking a series of object-related questions. Shen *et al.* [22] learned to caption images through a lifetime by generating caption-related questions. A decision-maker is introduced to learn when to ask questions by implicitly reasoning about the uncertainty of the agent and expertise of the teacher. Similar to the traditional VDN tasks [26, 18, 35, 20], these developments also suffer inflexible communication and expensive annotation cost. In contrast, our SCoA differs from these

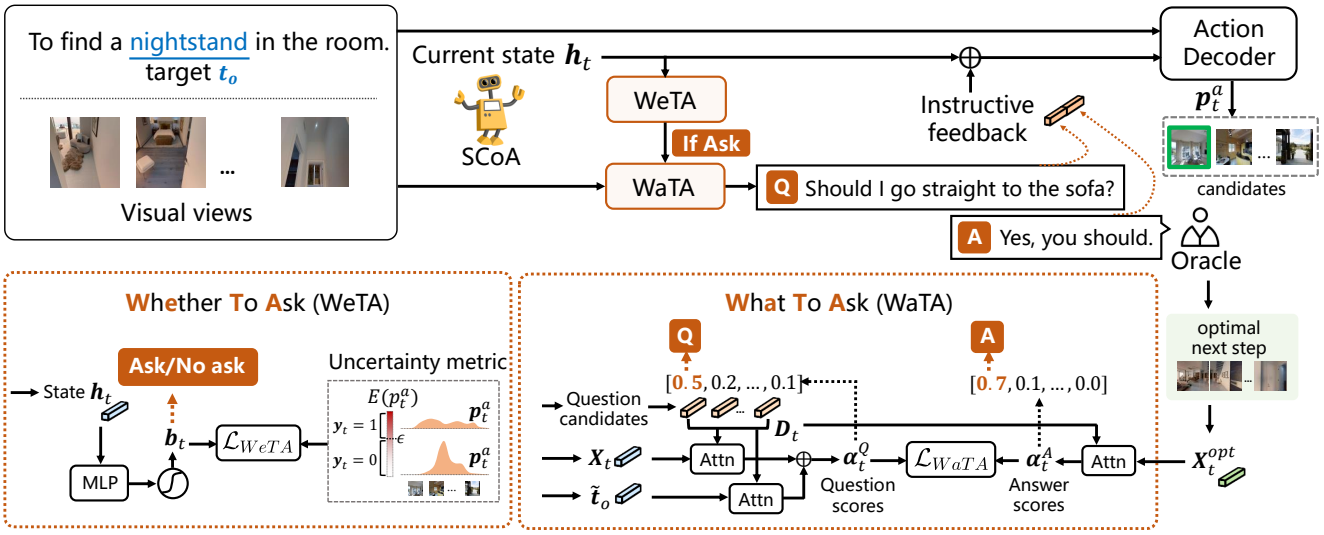

Figure 2: Training overview of our SCoA. At each step, WeTA learns to predict a probability b_t of asking, supervised by the entropy of action distribution p_t^a measuring the agent's uncertainty of which action to take. If asking, WaTA firstly generates question candidates, and then learns question scores α_t^Q for these candidates considering the language embedding $\tilde{t}_o$ and the vision features X_t. Besides, the answer scores α_t^A for the questions provided by the oracle who observes the optimal next step, are used to guide α_t^Q. As such, our agent can navigate in a self-Q&A manner even the oracle is unavailable in real world.

methods in not only the task domain, but also the communication scheme where whether and what to communicate with the oracle are self-adaptive without the involvement of labor-intensive dialog annotations.

3. Methodology

3.1. Preliminaries

Problem Definition. Given a house scan and a starting position, the agent is required to find a target t_o in a goal region by communicating with the oracle when it gets confused and has no idea what to do. The agent needs to decide whether and what to ask without any question annotations in this paper. We encode the target t_o via the word embedding to get feature $\tilde{t}_o \in \mathbb{R}^{d_w}$ and d_w is set to 300. At the t-th step, the agent receives a panoramic view which is then divided into $N = 36$ sub-images according to their camera angles of heading and elevation. Each image is represented by a feature vector $x_{t,i} \in \mathbb{R}^{d_f}$ extracted from the pre-trained Resnet-152 [6], appended with an embedding about heading and elevation of the camera. The d_f is set to 2048. The whole feature set is denoted as $X_t = \{x_{t,i}\}_{i=1}^{N} \in \mathbb{R}^{N \times d_f}$. The agent predicts an action a_t from an action set $\mathcal{A}_t$, which consists of current navigable viewpoints. Besides, the navigable viewpoint feature set is denoted as $Z_t = \{Z_{t,i}\}_{i=1}^{|\mathcal{A}|} \in \mathbb{R}^{|\mathcal{A}_t| \times d_f}$, also extracted from the ResNet-152.

Learning Framework. The VDN problem in existing researches [26, 35, 20] is often optimized with the reinforcement learning where only the navigation action prediction is organized as a policy π_μ (see Sec. 3.3). To support our motive of self-motivated communication to realize dialogue annotation-free navigation in real-world unseen environment, we further introduce a whether-to-ask (WeTA) policy π_ϕ (see Sec. 3.2.1) and a what-to-ask (WaTA) policy π_κ (see Sec. 3.2.2). As shown in Fig. 2, SCoA first decides whether to ask for help via WeTA policy when unsure where to go. If asking, the agent generates a question to ask via WaTA policy. During training, the oracle distills its knowledge about the optimal next step to guide the agent to mimic oracle's answering. This enables our agent to navigate in a self-Q&A manner even when the oracle is unavailable in real-world applications. The communication is jointly optimized with navigation in a unified imitation learning and reinforcement learning framework, driving the agent to reach target position with less communication cost (see Sec. 3.4).

3.2. Self-Motivated Communication Agent

The agents in existing works [26, 18, 35, 20] are only allowed to ask questions at predefined locations, leading to not only labor-intensive learning cost, but also inflexible human-robot communication in real-world application. To allow the agent to adaptively decide whether and what to ask, we propose a self-motivated communication scheme by introducing two policies of whether to ask and what to ask.

3.2.1 Whether to Ask

To fit the real-world application, the agent should be committed to adaptively decide whether to ask a question during

navigation rather than manually pre-defined. To that effect, at the t-th step, our agent learns to predict the probability of asking a question, $\boldsymbol{b}_t \in \{0, 1\}$, based on its current state $\boldsymbol{h}_t$ with a whether-to-ask policy $\pi_\phi(\boldsymbol{b}_t|\boldsymbol{h}_t)$[2], which is constructed by a Multi-Layer Perceptron (MLP), followed by the Gumbel-Softmax (GS) [4, 7]. The $\boldsymbol{b}_t$ is formulated as:

$$\boldsymbol{b}_t = \mathrm{GS}\big(\mathrm{MLP}(\boldsymbol{h}_t)\big). \tag{1}$$

During navigation, the agent needs to choose an action so as to move towards the target t_o. We denote the action probability distribution at the t-th step as $\boldsymbol{p}_t^a$. It is intuitive that the decision is hard to be made if $\boldsymbol{p}_t^a$ tends to be a uniform distribution where each action has the same probability to be chosen, which on the contrary brings more uncertainty. In this case, the agent will get confused and need auxiliary information to choose a wise action. This inspires us that $\boldsymbol{p}_t^a$ can be a valuable hint to guide the learning of our whether-to-ask policy π_ϕ. To this end, we introduce the entropy as a metric [18] to model the uncertainty of $\boldsymbol{p}_t^a$, and define the pseudo label $\boldsymbol{y}_t$ to supervise the whether-to-ask policy π_ϕ:

$$\boldsymbol{y}_t = \mathrm{one\_hot}\big(\,[\,\mathrm{H}(\boldsymbol{p}_t^a) < \epsilon\,]_+\,\big), \tag{2}$$

where $[\cdot]_+$ returns 1 if the condition of its input satisfies, and 0 otherwise; $\mathrm{H}(\cdot)$ returns the entropy of its input and $\epsilon \in [0, 1]$ is a pre-defined threshold. The motivation behind this is that a high entropy indicates that $\boldsymbol{p}_t^a$ would be more close to the uniform distribution. As such, the agent is considered to be uncertain about which action to choose, and thus needs to communicate with the oracle for help. Then, the learning of our whether-to-ask policy π_ϕ is regularized via the cross-entropy loss between $\boldsymbol{b}_t$ and $\boldsymbol{y}_t$:

$$\arg\min_{\pi_\phi} \mathcal{L}_{WeTA}(\boldsymbol{b}_t, \boldsymbol{y}_t; \pi_\phi) = -\mathbb{E}_{\boldsymbol{y}_t}\big[\log \boldsymbol{b}_t\big]. \tag{3}$$

3.2.2 What to Ask

Our SCoA learns to adaptively decide not only whether to ask, but also what to ask. To that effect, the WaTA first generates a question candidate set on-the-fly, and then picks up the most beneficial question to ask, which significantly differentiates our SCoA from existing works [26, 35, 20] where the question annotations are manually given in advance.

We train an encoder-decoder model [29] which generates a question for each image patch in the current panoramic view. The encoder takes as its input the image patch feature $\boldsymbol{x}_{t,i} \in \boldsymbol{X}_t$ associated with two keywords including "[Obj]" for object label and "[Dir]" for object location. Note that "[Obj]" and "[Dir]" are detected by an object localization network [32]. The decoder produces a question set $C_t = \{c_{t,i}\}_{i=1}^N$ for the sub-images in the panoramic view at the t-th navigation step. To this end, instead of resorting to the expensive dialog annotations, we train the encoder-decoder model using a small set of direction-related sentences, which are collected by filling the detected keywords into question templates (*e.g.*,"Should I go [Dir] to the [Obj]?") widely-used for asking directions[3]. Then, as shown in Fig. 2, we encode C_t via a word embedding layer, followed by a one-layer LSTM to generate question features $\boldsymbol{D}_t = \{\boldsymbol{d}_{t,1}, ..., \boldsymbol{d}_{t,N}\} \in \mathbb{R}^{d_l \times N}$, and d_l is set to 512.

We implement the policy $\pi_\kappa(\boldsymbol{\alpha}_t^Q|\widetilde{\boldsymbol{t}}_o, \boldsymbol{X}_t, \boldsymbol{D}_t)$ considering both language and vision information. Specifically, $\boldsymbol{\alpha}_t^Q \in \mathbb{R}^N$ is a question score vector measuring the importance of each question feature $\boldsymbol{d}_{t,i} \in \boldsymbol{D}_t$ from two aspects: (1) language information which measures the correlation between the question candidates and the target embedding $\widetilde{\boldsymbol{t}}_o$, and (2) vision information which measures the correlation between the question candidates and view features $\boldsymbol{x}_{t,i} \in \boldsymbol{X}_t$. We define the $\boldsymbol{\alpha}_t^Q$ as:

$$\boldsymbol{\alpha}_t^Q = \sigma\big(\underbrace{\sigma\big(\boldsymbol{D}_t(\widetilde{\boldsymbol{t}}_o \boldsymbol{W}^l)^T\big)}_{\text{Language information}} + \overbrace{\sigma\big(\sum_i \boldsymbol{D}_t(\boldsymbol{x}_{t,i}\boldsymbol{W}^v)^T\big)}^{\text{Vision information}}\big), \tag{4}$$

where $\sigma(\cdot)$ represents the softmax function, $\boldsymbol{W}^l \in \mathbb{R}^{d_w \times d_l}$ and $\boldsymbol{W}^v \in \mathbb{R}^{d_f \times d_l}$ are learnable weights.

Besides, we also introduce an answer score $\boldsymbol{\alpha}_t^A \in \mathbb{R}^N$ measuring the confidence of the oracle giving a positive answer embedding $\boldsymbol{s}_{t,i} \in \mathbb{R}^{d_l}$ (*e.g.*, "Yes, you should.") to each question $\boldsymbol{d}_{t,i} \in \boldsymbol{D}_t$. Specifically, we measure the correlation between the question candidates and the image features $\boldsymbol{X}_t^{opt} = \{\boldsymbol{x}_{t,i}^{opt}\}_{i=1}^N \in \mathbb{R}^{N \times d_f}$ of the optimal panoramic view at the next step, given by the oracle to compute $\boldsymbol{\alpha}_t^A$ as:

$$\boldsymbol{\alpha}_t^A = \sigma(\sum_{i=1}^N \boldsymbol{D}_t(\boldsymbol{x}_{t,i}^{opt}\boldsymbol{W}^a)^T), \tag{5}$$

where $\boldsymbol{W}^a \in \mathbb{R}^{d_f \times d_l}$ is the trainable weights.

The $\boldsymbol{\alpha}_t^A$ indeed can be viewed as knowledge of the future step from the oracle's observation. Thus, we propose to distill $\boldsymbol{\alpha}_t^A$ to assist the learning of question score $\boldsymbol{\alpha}_t^Q$ for the agent using the KL-divergence as:

$$\begin{aligned}\arg\min_{\pi_\kappa} &\mathcal{L}_{WaTA}(\boldsymbol{\alpha}_t^Q, \boldsymbol{\alpha}_t^A; \pi_\kappa) \\ &= \mathbb{E}_{\boldsymbol{\alpha}_t^Q}\big[\log \boldsymbol{\alpha}_t^Q\big] - \mathbb{E}_{\boldsymbol{\alpha}_t^Q}\big[\log \boldsymbol{\alpha}_t^A\big].\end{aligned} \tag{6}$$

The insights of our distillation are two-fold: First, during training, the oracle, who has a great store of knowledge about the optimal future step, offers affirmative responses, acting as a teacher to guide the agent to score the questions. Second, by optimizing the KL-divergence, the difference between the

[2]When $t = 0$, the state $\boldsymbol{h}_0$ is initialized by the target feature $\widetilde{\boldsymbol{t}}_o$.

[3]Please refer to the supplementary material for more details.

question score and answer score is minimized. Therefore, $\boldsymbol{\alpha}_t^Q$ plays as not only the question score, but also reflects the confidence of a "Yes" answer to the corresponding question. Through this, our agent can navigate in a self-Q&A manner without the involvement of the oracle in real world.

3.3. Where to Go

Our self-motivated communication enables the agent to adaptively decide whether and what to ask when communicating with the oracle. The next lies in which action the agent should take (where to go) so as to navigate towards to target with the policy $\pi_\mu(a_t|\boldsymbol{h}_t, a_{t-1}, \boldsymbol{X}_t, \boldsymbol{Z}_t)$.

As distinguished from existing works, our action prediction first takes into consideration the highest-score question feature and the corresponding answer embedding for the agent to update its current state $\boldsymbol{h}_t$ as:

$$\begin{aligned} &\boldsymbol{h}_t \leftarrow \boldsymbol{h}_t + \boldsymbol{b}_{t,0} \cdot [\boldsymbol{d}_{t,i}; \boldsymbol{s}_{t,i}]\boldsymbol{W}^d, \\ &s.t. \quad i = \arg\max_i \boldsymbol{\alpha}_{t,i}^Q , \end{aligned} \tag{7}$$

where $[\cdot\ ;\ \cdot]$ is the concatenation operation, and $\boldsymbol{W}^d \in \mathbb{R}^{2d_l \times d_l}$ is learnable weights. The rationale behind this is that the question score vector $\boldsymbol{\alpha}_t^Q$ constructed in Sec. 3.2.2 reflects the relative importance between questions by merging the language and vision information. Thus, in comparison with others, the question endowed with the highest score, can receive the most positive response (*e.g.*, "Yes") and is the most informative for moving towards the target.

We compute the attention $\widetilde{\boldsymbol{X}}_t$ between $\boldsymbol{h}_t$ and $\boldsymbol{X}_t$, which, along with previous action a_{t-1} and $\boldsymbol{h}_t$, is regarded as the input of a LSTM model to further update the state $\boldsymbol{h}_t$ as $\boldsymbol{h}_t \leftarrow \text{LSTM}([\widetilde{\boldsymbol{X}}_t; a_{t-1}], \boldsymbol{h}_t)$. The action a_t is predicted from a softmax function with the navigable viewpoint feature set $\boldsymbol{Z}_t$ and the updated state $\boldsymbol{h}_t$ as:

$$\boldsymbol{p}_t^a = \sigma(\boldsymbol{Z}_t \boldsymbol{W}^p \boldsymbol{h}_t), \tag{8}$$

where $\boldsymbol{W}^p \in \mathbb{R}^{d_f \times d_l}$ are trainable weights. Then, the action $a_t \in \mathcal{A}_t$ is sampled following the probability distribution $\boldsymbol{p}_t^a$. The objective function of the action decoder is defined as:

$$\arg\min_{\pi_\mu} \mathcal{L}_{Nav}(\boldsymbol{p}_t^a, \boldsymbol{p}_t^{a^*}; \pi_\mu) = -\mathbb{E}_{\boldsymbol{p}_t^{a^*}}\big[\log \boldsymbol{p}_t^a\big], \tag{9}$$

where $\boldsymbol{p}_t^{a^*}$ is a one-hot vector denoting the teacher's actions.

3.4. Model Optimization

In this section, we detail our model optimization for learning the self-motivated agent including the imitation learning which learns to imitate the behavior of given teachers, and the reinforcement learning which overcomes the misleading actions towards the teachers [25], as shown in Fig. 3.

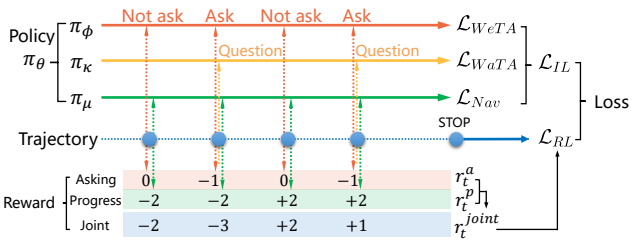

Figure 3: Optimization of our SCoA. The imitation learning ($\mathcal{L}_{IL}$) decides whether to ask, what to ask and where to go at each navigation step. The reinforcement learning ($\mathcal{L}_{RL}$) drives the agent towards the target position with less communication cost. (Best view in color)

Imitation Learning. The agent is trained with imitation learning to mimic the behaviors suggested by the uncertainty score $\boldsymbol{y}_t$ which denotes whether to ask, oracle's answer $\boldsymbol{\alpha}_t^A$ which denotes what to ask, and teacher's action $\boldsymbol{p}_t^{a^*}$ which denotes where to go. To this end, our objective for imitation learning is defined as:

$$\arg\min_{\pi_\theta} \mathcal{L}_{IL}\big((\boldsymbol{b}_t, \boldsymbol{y}_t), (\boldsymbol{\alpha}_t^Q, \boldsymbol{\alpha}_t^A), (\boldsymbol{p}_t^a, \boldsymbol{p}_t^{a^*}); \pi_\theta\big), \tag{10}$$

where $\pi_\theta = \{\pi_\phi, \pi_\kappa, \pi_\mu\}$, $\mathcal{L}_{IL} = \sum_t \mathcal{L}_{WeTA}(\boldsymbol{b}_t, \boldsymbol{y}_t; \pi_\phi) + \mathcal{L}_{WaTA}(\boldsymbol{\alpha}_t^Q, \boldsymbol{\alpha}_t^A; \pi_\kappa) + \mathcal{L}_{Nav}(\boldsymbol{p}_t^a, \boldsymbol{p}_t^{a^*}; \pi_\mu)$. Our imitation learning considers both the communication and navigation, the optimization of which helps the agent learn whether to ask, what to ask and where to go, so as to move towards the target, eventually.

Reinforcement Learning. We implement the on-policy reinforcement learning using Actor-Critic algorithm [17]. The actor is the policy π_θ with the parameters θ, which conducts actions in an environment. The critic computes state values V^{π_θ} to help assist the actor in learning. Besides, we introduce two types of rewards for the optimization of our policy models. At the t-th step, we assign policies π_κ and π_μ one progress reward r_t^p following [25], and policy π_ϕ a penalty (negative reward) r_t^a which constrains the frequency of asking questions. Specifically, r_t^a is assigned with -1 when the agent decides to ask, and 0 otherwise; r_t^p is assigned with either +2 if the agent closes to the target, or -2, otherwise. Thus, we can obtain the joint reward as $r_t^{joint} = r_t^p + r_t^a$, which has four states (-3, -2, +1 and +2) at each navigation step as illustrated in Fig. 3.

Then, given the state-action-reward (h_t, a_t, r_t^{joint}) of the observation at the t-th step, the A2C algorithm computes the accumulated reward $R_t^{joint} = \sum_{i=t}^{T} \gamma^{i-t}(r_t^{joint}) + \gamma^{T-t} V(\boldsymbol{h}_{T+1})$, where $\gamma \in [0, 1)$ is the discount factor and T is the maximum number of navigation actions. To obtain higher rewards, the agent explores to predict correct navigation actions while raising fewer questions in our framework. The objective for reinforcement learning is defined as:

$$\arg\min_{\pi_\theta} \mathcal{L}_{RL}(a_t, \boldsymbol{p}_t, R_t^{joint}, \boldsymbol{h}_t; \pi_\theta), \tag{11}$$

where $\mathcal{L}_{RL} = -\sum_t a_t log(\boldsymbol{p}_t)(\mathcal{R}_t^{joint} - V^{\pi_\theta}(\boldsymbol{h}_t)) + \lambda_{RL}\sum_t(\mathcal{R}_t^{joint} + \gamma V^{\pi_\theta}(\boldsymbol{h}_{t+1}) - V^{\pi_\theta}(\boldsymbol{h}_t))^2$, and λ_{RL} is the weight for balancing the actor and critic. Finally, the overall objective for our SCoA can be formulated as:

$$\arg\min_{\pi_\theta} \mathcal{L}_{RL} + \mathcal{L}_{IL}. \tag{12}$$

4. Experiment

4.1. Settings

Datasets. We evaluate our SCoA on the CVDN [26] and REVERIE [19]. CVDN contains 2,050 human-human navigation instances across 83 MatterPort houses. These instances are further split up into 7k shorter navigation instances, including 4,742 instances for training, 382 instances for seen validation, 907 instances for unseen validation, and the others for test. Besides, three types of paths are provided in CVDN including: (1) Navigator paths which are annotated by humans; (2) Oracle paths which denote the shortest paths; (3) Mixed paths which consist of either the navigator path if the end nodes of the navigator and oracle are the same, and the oracle path, otherwise. We implement most experiments using 7k shorter navigation instances except for Tab. 2 with 2,050 navigation. As for REVERIE, it has 21,702 instructions which are then partitioned into: a training set with 10,466 instructions over 2,353 objects, a seen validation set with 4,944 instructions over 953 objects, and an unseen validation set with 3,573 instructions over 525 objects.

Metrics. Four metrics are used, including: (1) Goal Progress (GP), which indicates the average progress towards the target; (2) Success Rate (SR), which denotes the percentage of reaching the position within three meters of the target. (3) Oracle Success Rate (OSR), which represents the percentage of reaching the position closest to the target. (4) Success rate weighted by Path Length (SPL).

Implementation Details. Our model is implemented using Pytorch and trained on a Tesla P100 for 20,000 iterations. For the planner path, the maximum step number T and the batch size are set to 20 and 80; otherwise, we set T to 80 and batch size to 40. The Adam optimizer [9] with learning rate of 0.0001 is adopted for updating.

4.2. Ablation Study

This part of experiments mainly focuses on evaluating the effects of our whether-to-ask module (WeTA) and what-to-ask module (WaTA), respectively. Without loss of generality, all experiments are conducted on CVDN using the oracle paths and we report the performance *w.r.t.* the Goal Progress (GP). Note that, we report the performance of our SCoA based on a self-Q&A manner where the oracle is removed during testing to mimic the real-world environment.

Effect of WeTA. We start with the analysis of the whether-to-ask module. To that effect, we compare our agent against non-learning agents and learning agents. Non-learning agents: (1) Never: the agent never communicates with the oracle. (2) Random: the oracle chooses to request help with a probability of 0.4, which is also the statistical result of our SCoA. (3) Always: the oracle requests help at each navigation step. Learning agents: (1) IC3Net [24]: uses the softmax layer to indicate whether to ask. (2) When2com [13]: requests help based on the correlation between the key and query transformed from the observations. Tab. 1 shows our experimental results. Our WeTA significantly outperforms the other methods in both seen and unseen environments. Besides, comparing to the non-learning "Always", WeTA asks questions if necessary, which requires less communication cost. Comparing to When2com [13] where a complex multi-agent perception system has to be built, our WeTA merits in its simplicity yet effectiveness.

Method	Val Seen (m)	Val Unseen (m)
Non-learning Agent		
Never	4.1	1.73
Random	5.03	1.74
Always	5.41	1.78
Learning Agent		
IC3Net [24]	4.83	1.76
When2com [13]	4.88	1.86
SCoA (**Ours**)	**5.93**	**1.94**

Table 1: We replace the whether-to-ask (WeTA) module in our SCoA with existing (non-)learning methods to analyze its effectiveness. Following [24, 13], we show the Goal Progress (m) on CVDN using the oracle path.

Method	Val Seen (m)	Val Unseen (m)
Shortest Path	32.8	29.3
RMM$_{N=3}$ [20]	14.0	5.6
RMM$_{N=3}$ + Oracle Stopping [20]	16.8	8.9
SCoA (**Ours**)	**19.52**	**11.19**

Table 2: We replace the what-to-ask (WaTA) module in our SCoA with existing question generation mechanism [20] to analyze its effectiveness. Following [20], we show the Goal Progress (m) on CVDN with 2,050 navigation instances using the mixed path.

Effect of WaTA. Then, we analyze the effects of our what-to-ask module from two aspects: the question generation mechanism and the selected questions. Tab. 2 shows comparison between existing question generation mechanism [20] and our WaTA. The "shortest path" plays as the upper bound of expected performance. As can be observed, compared with RMM [20] which trains an encoder-decoder

Method	Dialog Annotation	Val Seen			Val Unseen			Test Unseen		
		Oracle	Navigator	Mixed	Oracle	Navigator	Mixed	Oracle	Navigator	Mixed
Shortest Path	✗	8.29	7.63	9.52	8.36	7.99	9.58	8.06	8.48	9.76
Random	✗	0.42	0.42	0.42	1.09	1.09	1.09	0.83	0.83	0.83
Vision Only	✗	4.12	5.58	5.72	0.85	1.38	1.15	0.99	1.56	1.74
SCoA (**Ours**)	✗	**5.93**	**6.70**	**7.11**	**1.94**	**2.91**	**2.85**	**2.49**	**3.37**	**3.31**
Seq2seq [26]	✓	4.48	5.67	5.92	1.23	1.98	2.10	1.25	2.11	2.35
CMN [35]	✓	5.47	6.14	7.05	2.68	2.28	2.97	2.69	2.26	2.95
PREVALEN [5]	✓	-	-	-	2.58	2.99	3.15	1.67	2.39	2.44

Table 3: Performance comparison between our SCoA and existing methods. We show the Goal Progress (m) on CVDN using the three provided types of paths.

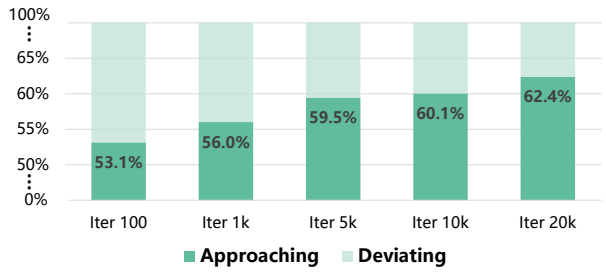

Figure 4: Effect of selected questions by WaTA. We show the percentages of selected questions that guide the agent to approach to (deviate from) the target at different training iterations (seen validation on CVDN using the oracle path).

framework to generate questions using dialogue annotations, our WaTA yields a supreme performance even when the oracle stopping [20] is combined with RMM. This well demonstrates the feasibility and effectiveness of our question generation using a small set of direction-related sentences.

To analyze the effectiveness of each selected question, we calculate the percentages of questions that guide the agent to approach (deviate from) the target. As shown in Fig. 4, the percentage of "approaching" goes up as the network training, denoting that our agent learns to pick up beneficial questions. Finally, a high percentage of 62.4% is derived, showing that our agent has a strong ability to distinguish informative question for moving towards the target.

Effect of Loss Regularization. In Sec. 3.2.1 and Sec. 3.2.2, we introduce the cross-entropy loss $\mathcal{L}_{WeTA}$ and KL-divergence $\mathcal{L}_{WaTA}$ to learn the modules of whether to ask and what to ask. In Tab. 4, we respectively remove one of them and show the performance. As can be seen, the performance of SCoA drops significantly without $\mathcal{L}_{WeTA}$ or $\mathcal{L}_{WaTA}$, which well demonstrates the importance of these two loss constraints in learning our self-motivated agent.

4.3. Performance Analysis

In this subsection, we conduct an experimental comparison with existing VDN methods on CVDN and REVERIE. Then, we dive into an in-depth insight into how the communication and navigation in our SCoA are mutually optimized.

Method	Val Seen (m)	Val Unseen (m)
SCoA w/o $\mathcal{L}_{WeTA}$	5.49	1.87
SCoA w/o $\mathcal{L}_{WaTA}$	5.78	1.86
SCoA (**Ours**)	**5.93**	**1.94**

Table 4: Performance analyses without $\mathcal{L}_{WeTA}$ and $\mathcal{L}_{WaTA}$. We show the Goal Progress (m) on CVDN using oracle path.

Results on CVDN. We first build three baseline methods: (1) The Shortest Path Agent takes the shortest path to the goal which denotes the upper bound of navigation performance. (2) The Random Agent chooses a random heading and moves 5 steps forward each time. (3) The Vision Only agent ignores language input. Besides, three existing competitors with rich dialog annotations, including Seq2seq [26], CMN [35] and PREVALEN [5], are introduced for comparison. In Tab. 3, our SCoA shows an overwhelming superiority over the baselines without dialog annotations on both seen and unseen environments over different types of paths. Particularly, SCoA even shows comparable results to these with dialog annotations as inputs, which well demonstrates the ability of SCoA to generate informative questions for navigation. Furthermore, we add the oracle during testing, and the results *w.r.t.* oracle path, navigator path and mixed path further increase to 6.74, 7.00 and 8.02 on seen validation, and 2.30, 2.64 and 3.28 on unseen validation.

Results on REVERIE. Tab. 5 shows the performance comparison on REVERIE. The compared methods take the annotated instructions as inputs on both the training set and validation set, while our SCoA gets rid of the instructions. Instead, it generates dialogs on-the-fly. As can be seen, compared with the recent advance [19], our SCoA performs the best in all three evaluation metrics.

Progressive-Suppressive Learning. In Fig. 5, we detailedly analyze the learning of our SCoA by counting the number of asking questions and the gained joint reward at different navigation steps. We observe that our SCoA learns to

Method	Val Unseen		
	SR ↑	OSR ↑	SPL ↑
Random	1.76	11.93	1.01
R2R Teacher Forcing [1]	3.21	4.94	2.80
R2R Student Forcing [1]	12.88	4.20	8.07
RCM [27]	9.29	14.23	6.97
Self-Monitor [15]	8.15	11.28	6.44
FAST-Short [8]	10.08	20.48	6.17
Navigator-Pointer [19]	14.40	28.20	7.19
SCoA (**Ours**)	**16.94**	**29.29**	**8.2**

Table 5: Performance comparison on the unseen validation of the REVERIE. Three metrics, including SR (%), OSR (%) and SPL (%), are introduced.

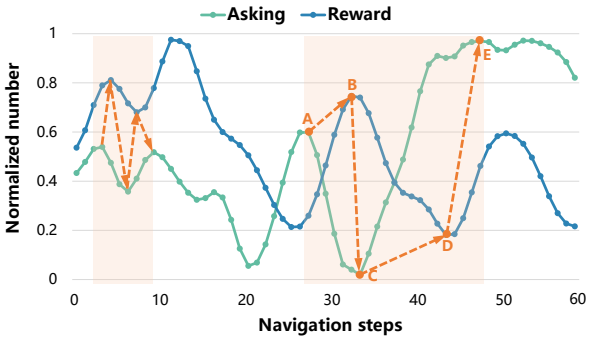

Figure 5: The normalized number of asking questions *v.s.* the joint reward at different navigation steps on the unseen validation of CVDN during inference using the mixed path.

navigate in a progressive-suppressive manner. Specifically, when our agent gets stuck in where to go, it tends to ask more questions (*e.g.*, point A). This results in the increase of reward (*e.g.*, point B), which indicates that our agent is progressively approaching to the target. However, the increase of reward on the contrary suppresses the agent to ask questions (*e.g.*, point C) to reduce the communication cost since it has received rich knowledge about the surroundings of its current position. However, as the agent goes beyond the surroundings, more auxiliary information is necessary to support the ongoing exploration. Otherwise, the agent would pick up wrong actions, resulting in decreasing reward (*e.g.*, point D). To take back the right direction, the agent again resorts to asking more questions (*e.g.*, point E). Thus, our SCoA implements the progressive-suppressive learning in a closed loop until the agent reaches the target with less communication cost.

Trajectory Visualization. We visualize one trajectory example of our SCoA in Fig. 6 to see how our SCoA performs the VDN task, along with the highest-score questions and the joint rewards (see Fig. 3). As can be observed, the agent receives negative rewards when it deviates from the

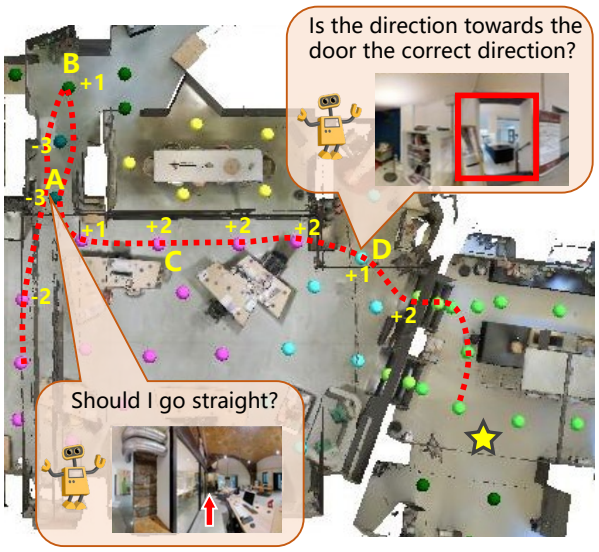

Figure 6: Trajectory visualization of our SCoA. The red dotted line represents the trajectories the agent has traveled, and the yellow star indicates the target position. The rectangle box contains the asked questions by the agent and their corresponding scene images. The digits denote the joint rewards at each step. (Best view in color)

target position. Particularly, the agent gets confused at point A, and then asks a question of "Should I go straight?", which however still leads the agent to a wrong direction thus a penalty of -3 is received. The agent keeps asking questions until choosing a right action at point B which returns a reward of +1. Then, it stops communicating with the oracle at point C since the agent knows the surroundings and starts to move towards the target, during which, a reward of +2 is received. The ongoing moving leads the agent out of the surrounds of point C. Thus, the agent asks one more question of "Is the direction towards the door the correct direction" at point D for its further exploration.

5. Conclusion

In this paper, we propose the Self-Motivated Communication Agent (SCoA) to tackle the challenging problem of inflexible and annotation-dependent communication for real-world vision-dialog navigation by learning to adaptively decide whether and what to communicate with human to acquire instructive information for guiding the navigation. By jointly learning to communicate and navigate, SCoA explores to balance the communication benefit and cost. SCoA significantly outperforms existing baseline methods without dialog annotations, and even achieves comparable performance to the counterparts that use rich dialog annotations as inputs. Our SCoA gets rid of the limitation of expensive language annotations and shows great potential for navigating in real and open-ended environments.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3674–3683, 2018. 1, 2, 8

[2] Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5503–5512, 2017. 2

[3] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 3318–3329, 2018. 1, 2

[4] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. 1954. 4

[5] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13137–13146, 2020. 2, 7

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[7] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[8] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6741–6749, 2019. 8

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[10] Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Learning through dialogue interactions by asking questions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 2

[11] Yimeng Li and Jana Košecka. Learning view and target invariant visual servoing for navigation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 658–664, 2020. 2

[12] Bingqian Lin, Yi Zhu, Yanxin Long, Xiaodan Liang, Qixiang Ye, and Liang Lin. Retreat for advancing: Dynamic reinforced instruction attacker for robust visual navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 2

[13] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4106–4115, 2020. 6

[14] Yunlian Lv, Ning Xie, Yimin Shi, Zijiao Wang, and Heng Tao Shen. Improving target-driven visual navigation with attention on 3d spatial relationships. *arXiv preprint arXiv:2005.02153*, 2020. 2

[15] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019. 8

[16] Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens Van Der Maaten. Learning by asking questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2018. 2

[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016. 5

[18] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, 2019. 1, 2, 3, 4

[19] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9982–9991, 2020. 6, 7, 8

[20] Homero Roman Roman, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao. Rmm: A recursive mental model for dialog navigation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1732–1745, 2020. 1, 2, 3, 4, 6, 7

[21] Mrinmaya Sachan and Eric Xing. Self-training for jointly learning to ask and answer questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 629–640, 2018. 2

[22] Tingke Shen, Amlan Kar, and Sanja Fidler. Learning to caption images through a lifetime by asking questions. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 10393–10402, 2019. 2

[23] William B Shen, Danfei Xu, Yuke Zhu, Leonidas J Guibas, Li Fei-Fei, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2881–2890, 2019. 2

[24] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018. 6

[25] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2610–2621, 2019. 2, 5

[26] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 394–406, 2020. 1, 2, 3, 4, 6, 7

[27] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei 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 (CVPR)*, pages 6629–6638, 2019. 1, 2, 8

[28] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 2

[29] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International conference on machine learning (ICML)*, pages 2048–2057. PMLR, 2015. 4

[30] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Visual curiosity: Learning to ask questions to learn visual recognition. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR, 2018. 2

[31] Xin Ye and Yezhou Yang. Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling. *arXiv preprint arXiv:2010.08596*, 2020. 2

[32] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. 4

[33] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

[34] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10012–10022, 2020. 1, 2

[35] Yi Zhu, Fengda Zhu, Zhaohuan Zhan, Bingqian Lin, Jianbin Jiao, Xiaojun Chang, and Xiaodan Liang. Vision-dialog navigation by exploring cross-modal memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10730–10739, 2020. 1, 2, 3, 4, 7