# Vision-Dialog Navigation and Beyond

**Zikai Wei**
Department of Information Engineering
The Chinese University of Hong Kong
1155111173
wz018@ie.cuhk.edu.hk

**Jiankai Sun**
Department of Information Engineering
The Chinese University of Hong Kong
1155136477
sj019@ie.cuhk.edu.hk

## Abstract

Vision-dialog navigation problem is an existing challenging. Many existing work has done on this problem using supervised learning methods. They extract the historical dialogues and visual information, and further convey those extracted information to a later classifier. The classifier later provides the agent action. However, reinforcement learning has not been implemented in this area. Our motivation is to use reinforcement learning algorithms to solve the vision-dialog navigation problem. We firstly create our reinforcement learning environment on the vision-and-dialog navigation dataset, Matterport3D. Then, we implement two reinforcement learning algorithms, TRPO and A2C, on our vision-dialog navigation environment. The empirical results show that reinforcement learning on vision-dialog navigation problem can obtain a similar results to the existing supervised learning methods (“rule-based methods based on the shortest path"). A demo video is available[1].

## 1 Introduction

Vision-dialog navigation problem becomes more and more [1, 10, 11]. Dialog assistants, which can communicate via natural language and entertain humans, have been widely adopted in recent research. These systems can carry information, without objects manipulation or actuator. On the order hand, mobile robots do not interact with human users but are still largely deployed in industrial settings. Besides, navigating from place to place successfully is a fundamental need for a robot in a home environment. The navigation can be facilitated, as with smart assistants, through dialog. To study this challenge, we want to use datasets such as Matterport3D to train navigation agents by asking targeted questions about where to go next when unsure. Vision-dialog navigation dataset in [10] provides a good playground for creating natural language interfaces helping robots and non-experts collaborate to achieve their goals.

In the previous works [1, 10, 11], they use the supervised learning to solve the vision-dialog navigation problem. They extract the historical dialogues and visual information, and further convey those extracted information to a later classifier. This classifier will decide the action an agent may take in the near future. More specifically, they use the sequence-to-sequence model to encode the human-to-human dialog. Meanwhile, there is another module processing the visual information, which may associate the vocal and visual information to an action decision. They try to get a closer location near to the target location with inferring navigation actions. They have shown that the agents can perform better with more dialog history and more supervision from humans and planners in the training phase.

However, the existing works do not use an reinforcement learning pipeline. Our **motivation** is to use reinforcement learning algorithms to solve the vision-dialog navigation problem. The existing work

[1] https://drive.google.com/file/d/1-sd51wnEdXp20udI3uZ3yoH-5Sj7sb0R/view?usp=sharing

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

uses supervised learning to solve this problem. In our work, we write a reinforcement learning environment for the vision-dialog navigation problem and implement Trust Region Policy Optimization (TRPO) [8] and Advantage Actor Critic (A2C) [6, 9] on this problem. We firstly create our reinforcement learning environment on the vision-and-dialog navigation dataset, Matterport3D [3, 10]. Then, we implement TRPO and A2C on our vision-dialog navigation environment. The results shows that our reinforcement learning pipeline can attain similar performance to the “rule-based" (supervised learning from the shorest path) method.

Our work has the following merits: 1) We create an reinforcement learning environment exclusive for Matterport3D dataset, which incorporates vision-dialog information. 2) We implement TRPO and A2C on our vision-dialog navigation environment and empirically shows the performance gap between the shortest-rule-based method and the methods using reinforcement learning. 3) Our empirical results shows that reinforcement learning could be one valuable orientation in vision-dialog navigation.

## 2 Related Work

Dialogs start with an underspecified, ambiguous instruction similar to what robots may encounter in a home environment (e.g., “Go to the pantry with the cook”). Dialogs include not only navigation but also question asking/answering to guide the search, which is similar to a robot agent conducting clarification when moving through a new environment.

**Vision-and-Language Navigation.** Early, simulator-based Vision-and-Language Navigation (VLN) tasks use language instructions that are unambiguous, designed to uniquely describe the goal, and fully specified describing the steps necessary to reach the goal [5]. In photo-realistic simulation environments, agents can navigate high-definition scans of indoor scenes or large, outdoor city spaces. In interactive question answering settings, the language context with a single question (e.g., “What color is the car?") requires navigation to answer. The questions serve as under specified instructions but are not clear (e.g., there is only one car, and what color of it can be asked about). One needs the questions generated from the human language. However, those questions are still generated from templates.

**Exploring Cross-modal Memory.** In vision-dialog navigation, an agent is expected to take advantage of a consecutive dialog with humans or oracles. How to make good use of the history of the past dialog and senses is necessary in vision-dialog navigation. Cross-modal memory [11] is try to understand and remember the vision-dialog information relevant to historical navigation actions. Cross-modal memory is implemented on two ways: one is the language memory module and the other is the visual memory module. In particular, the language memory try to learn the relationship between the historical dialog and the current conversation. The visual memory module learns to associate the current visual information and the cross-modal memory of the previous navigation actions.

**Question Answering and Dialog.** In Visual Question Answering (VQA), agents have to answer questions, given a static image as the extra info. These tasks are templated language on rendered images and human language on real-world images [2]. Later extensions feature two-sided dialog, where a series of question-answer pairs provide context for the next question. Question answering in natural language processing has been studied for a long time on the tasks with questions about static text documents (e.g., the Stanford QA Dataset). Recently, this paradigm was extended to dialogs from two sides, which is based on pairs of human-human and question-answer. Questions in these datasets are not clear: they need to infer from the context to obtain a correct answer.

**Task-oriented Dialog.** In human-robot interaction, robot language, with the help of humans, can be generalized to non-verbal human help. However, humans may give verbal responses to robot requests for help in task-oriented dialogs. A recent work view “the requesting navigation help" as an action, but the response can either comes from two forms: one is a templated language which can encode gold-standard planner action sequences; the other is an automatic generation. The generator is trained from human instructions and coupled with a visual goal frame as additional supervision. Past work introduced Talk the Walk (TtW), where two humans communicate to reach a goal location in an outdoor environment. In TtW, the guiding human only has an abstracted semantic map, without so much visual feature but with some semantic elements, e.g., “restaurant”. The target location is no clear to the guide in the beginning. In CVDN, a Navigator human requests for help with the generated language. Then, an Oracle human answers in vocal response conditioned on higher-level, visual observations of what a shortest-path planner would do next. Both sides can observe the same egocentric visual information.

# 3 Environment and Data: The Cooperative Vision-and-Dialog Navigation Dataset

From 83 MatterPort [3] houses, we collect some human-human navigation dialogs, comprising over about 7,000 navigation trajectories punctuated by question-answer exchanges. We prompt with initial instructions that are both ambiguous and underspecified. An ambiguous navigation instruction is one that requires clarification because it can refer to more than one possible goal location. An underspecified navigation instruction is one that does not describe the route to the goal.

**Dialog Prompts.** A dialog prompt is a tuple, which consists of the house scan $S$, a target object needed to be found, a starting position p0, and a goal region $G_j$. We suppose to use MatterPort object segmentations to get region locations for household objects, as in prior work [7]. We keep the same settings as defining “a set of 81 unique object types that appear no less than 5 distinct houses which appear between 2 and 4 times" [7]. Each dialog begins with an ambiguous and underspecified hint. One example given in the previous work is such as “The goal room contains a plant,” which by construction is both ambiguous (there are two to four rooms with a plant) and underspecified (the path to the room is not described by the hint).

# 4 Proposed Approach

## 4.1 Our environment

We create our reinforcement learning environment based on on the vision-and-dialog navigation dataset, Matterport3D [3, 10].

**States.** In our environment, we present the information in three aspctes:

- Questions asked by Navigator. Navigator tries to get to the destination. Navigator may need help during its navigation. Once Navigator needs extra information to help itself, it will raise a question from Oracle.
- Answers from Oracle. Oracle can be viewed as a human coach. Oracle will provide extra information help Navigator to find direction or a better path to its destination.
- Visual information from the surroundings. Navigator can also obtain the visual information from its current view of its surroundings.
- Current location. Navigator also knows where its current location is.

For the verbal information, we use sequential models to extract the language features. We also use ResNet to extract the image features.

**Reward function.** The reward function is defined as the topological distance the agent advanced.

$$r_t = d_t - d_{t-1},$$

where $d_t$ is **topological distance** between the agent’s current location and its destination. **Topological distance** has already consider the path nearby the wall and other obstacles, which means the agent cannot walk through a obstacle and the agent can only go around or avoid obstacles.

**Action.** The agent can take different actions to get to its destination.

$$a_t = \Pi(s_t),$$

where, $\Pi$ is the policy, $s_t$ is the current state, and $a_t$ is the agent’s action

$$a_t \in \{\text{forward, backward, left, right, up, down}\}.$$

### 4.2 Advantage Actor Critic

As a policy gradient algorithm, Advantage Actor Critic (A2C) [6] is part of the on-policy family.

the TD error is defined as

$$TD = r + \gamma V(s') - V(s) \tag{1}$$

The advantage function is defined as

$$\begin{aligned} A(s_t, a_t) &= Q_w(s_t, a_t) - V_v(s_t) \\ &= r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t) \end{aligned} \tag{2}$$

The update equation is

$$\begin{aligned} \nabla_\theta(J(\theta)) &\sim \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)(r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)) \\ &= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t) \end{aligned} \tag{3}$$

The critic loss is basically the MSE between TD target and current state value, as the difference on the advantage. For the actor, the actor loss is using the negative log likelihood, scaled by the advantage.

### 4.3 Trust Region Policy Optimization

Let $\pi_\theta$ denote a policy with parameters $\theta$. The theoretical TRPO [8] update is:

$$\begin{aligned} \theta_{k+1} &= \arg\max_\theta \mathcal{L}(\theta_k, \theta) \\ &\text{s.t. } \bar{D}_{KL}(\theta||\theta_k) \leq \delta \end{aligned} \tag{4}$$

where $\mathcal{L}(\theta_k, \theta)$ is the surrogate advantage, a measure of how policy $\pi_\theta$ performs relative to the old policy $\pi_{\theta_k}$ using data from the old policy:

$$\mathcal{L}(\theta_k, \theta) = E_{s,a\sim\pi_{\theta_k}} \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), \tag{5}$$

and $\bar{D}_{KL}(\theta||\theta_k)$ is an average KL-divergence between policies across states visited by the old policy:

$$\bar{D}_{KL}(\theta||\theta_k) = E_{s\sim\pi_{\theta_k}} D_{KL}\left(\pi_\theta(\cdot|s)||\pi_{\theta_k}(\cdot|s)\right). \tag{6}$$

The theoretical TRPO update isn't the easiest to work with, so TRPO makes some approximations to get an answer quickly. We Taylor expand the objective and constraint to leading order around $\theta_k$:

$$\begin{aligned} \mathcal{L}(\theta_k, \theta) &\approx g^T(\theta - \theta_k)\bar{D}_{KL}(\theta||\theta_k) \\ &\approx \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \end{aligned} \tag{7}$$

resulting in an approximate optimization problem,

$$\begin{aligned} \theta_{k+1} &= \arg\max_\theta g^T(\theta - \theta_k) \\ &\text{s.t. } \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \leq \delta. \end{aligned} \tag{8}$$

We divide all the data collected from Matterport3D [3] into training, validation, and test folds. We use the navigation steps taken by the Navigator after the question-answer exchange and the shortest-path steps shown to the main speaker and used as context to provide an answer.

In each instance of the task, the QA exchange in the dialog from which the instance is drawn (with $i = 0$ an empty QA followed by initial navigation steps). The steps range in length from 1 to 40. The Navigator often continues farther than what the main speaker describes, using their intuition about the house layout to seek the target object.

Figure 1: **Environment and Dataset.**

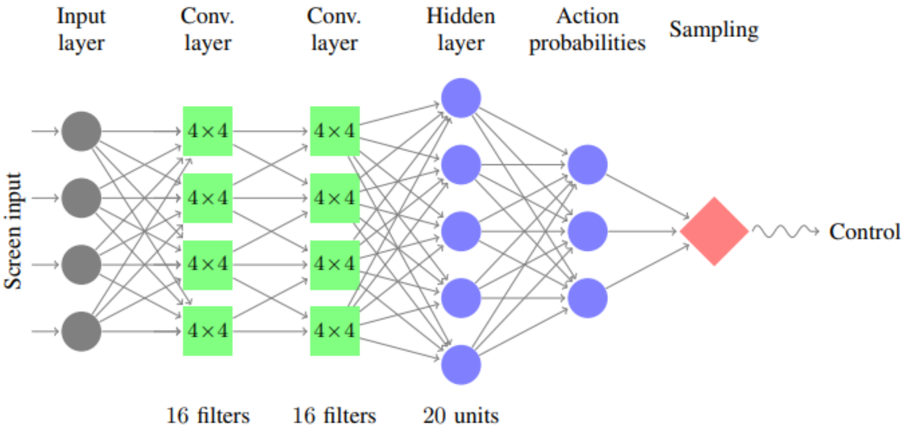

Figure 2: Neural Network for Reinforcement Learning

We evaluate the performance on the vision-dialog navigation task by measuring how much progress the agent makes towards goal region $G_j$. Let $e(P)$ be the end node of path $P$, $b(P)$ the beginning, and $\hat{P}$ the path inferred by the navigation agent. Then the progress towards the goal is defined as the reduction (in meters) from the distance to the goal region $G_j$ at $b(\hat{P})$ versus at $e(\hat{P})$.

Because $G_j$ is a set of nodes, we take the minimum distance $\min \mathbf{p} \in \mathbf{G_j}$ as the distance between $q$ and region $G_j$ . This is a topological distance. For example, we measure the distance around a wall, rather than straight through it).

## 5 Experiments

We designed our experiments to investigate the following questions: We conducted the Vision-Dialog Navigation experiments using Vision-Dialog Navigation environment, reformed from the CVDN simulator [10]. The simulated robots are shown in Figure 1. The states of the agents are their RBGD visual input, and the language embedding. For A2C, we used the general setting as shown in [6]. For TRPO, we used $\delta = 0.01$ for all experiments and used neural networks to represent the policy, with the architecture shown in Figure 2,

Anderson *et al.* [1] introduced a sequence-to-sequence model to serve as a learning baseline in the R2R task. A sequence of navigation instructions is used for an initial learning baseline for the Navigation task rather than a single navigation instruction.

Table 1: Comparison among A2C, TRPO, and existing baselines

| Agent Type | Val(Seen) | | | Val(Unseen) | | | Test(Unseen) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Oracle | Navigator | Mixed | Oracle | Navigator | Mixed | Oracle | Navigator | Mixed |
| Shortest Path Agent | 8.29 | 7.63 | 9.52 | 8.36 | 7.99 | 9.58 | 8.06 | 8.48 | 9.76 |
| Random Agent | 0.42 | 0.42 | 0.42 | 1.09 | 1.09 | 0.83 | 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 |
| Dialog Only | 1.41 | 1.43 | 1.58 | 1.68 | 1.39 | 1.64 | 1.51 | 1.20 | 1.40 |
| Seq2Seq | 4.48 | 5.67 | 5.92 | 1.23 | 1.98 | 2.10 | N/A | N/A | N/A |
| A2C | 3.24 | 3.76 | 4.01 | 0.85 | 1.05 | 1.10 | N/A | N/A | N/A |
| TRPO | 4.55 | 5.60 | 5.82 | 1.21 | 2.01 | 2.11 | N/A | N/A | N/A |

Table 2: Comparison among Seq2Seq, A2C, and TRPO

| Agent Type | Val(Seen) | | | | Val(Unseen) | | | |
|---|---|---|---|---|---|---|---|---|
| | SR | OSR | GP | OPSR | SR | OSR | (GP) | OPSR |
| Seq2Seq | 5.92 | 63.8 | 36.9 | 72.7 | 2.10 | 25.3 | 13.7 | 33.9 |
| A2C | 5.02 | 57.8 | 32.9 | 69.2 | 2.11 | 24.3 | 12.9 | 32.9 |
| TRPO | 6.01 | 64.2 | 37.4 | 73.2 | 2.16 | 28.0 | 12.8 | 36.7 |

With an LSTM, The dialog history is encoded, and the hidden state of an LSTM decoder is initialized with visual frames from the environment as observations. The outputs of the LSTM decoder are actions in the environment. We replace words that occur fewer than certain times. During the training phase, an embedding is given as input to the encoder, and the encoder learned for every token. We embed the visual frame using an Imagenet-pre-trained ResNet-152 model [4] for visual features.

We only train on the training fold. After that, we evaluate the validation folds. We want to introduce a mixed planner and human supervision strategy at training time and ablate the distance of dialog history encoded for ablation study.

As hypothesized, both that encoding a more extended dialog history and the use of mixed-supervision steps probably increase the amount the agent progresses towards the goal.

The experiment results are shown in Table 2 and Table 2. We use the same matrices in [1, 10, 11] are: Success Rate (SR), Oracle Success Rate (OSR), Goal Progress (GP), Oracle Path Success Rate (OPSR) , Success Rate (SR), Oracle Success Rate (OSR), Goal Progress (GP), and Oracle Path Success Rate (OPSR). The larger the value is, the better the performance is.

From the experiment results, we can see the Seq2seq [10], currently one of the best baselines, performance still better than our RL pipelines. However, our reinforcement implementations with A2C and TRPO can achieve better results in Success Rate (SR). This shows reinforcement learning can be a possibly profound orientation in vision-dialog navigation.

## 6 Conclusions and Future works

We create an reinforcement learning environment on the vision-and-dialog navigation dataset, Matterport3D. We then implement two reinforcement learning algorithms, TRPO and A2C, on our vision-dialog navigation environment. The empirical results show that reinforcement learning on vision-dialog navigation problem can obtain a similar results to the existing supervised learning methods:

- Seq2seq is still better than our RL pipelines (Goal progress)
- However, RL pipeline can somehow do well in the perspective of success rate.

**Future works.** In the future work, we will implement imitation learning as another comparison. We can revise our rewards function to an imitation loss

$$L^{IL} = \sum_t L_t^{IL} = \sum_t -log(p_t)a_t^*,$$

where $p_t$ is the probability distribution of agent's actions, and $a_t^*$ is the optimal action the agent shall take. Besides, we can also implement variational recurrent neural networks and graph structure to extract the verbal information from the dialog between Navigator and Oracle.

# 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*, pages 3674–3683, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[4] 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*, pages 770–778, 2016.

[5] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4, 2006.

[6] 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 *International conference on machine learning*, pages 1928–1937, 2016.

[7] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019.

[8] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[9] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[10] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *arXiv*, 2019.

[11] Yi Zhu, Fengda Zhu, Zhaohuan Zhan, Bingqian Lin, Jianbin Jiao, Xiaojun Chang, and Xiaodan Liang. Vision-dialog navigation by exploring cross-modal memory. *arXiv preprint arXiv:2003.06745*, 2020.