Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation

Xin Wang[1] Qiuyuan Huang[2] Asli Celikyilmaz[2] Jianfeng Gao[2] Dinghan Shen[3]
Yuan-Fang Wang[1] William Yang Wang[1] Lei Zhang[2]
[1]University of California, Santa Barbara [2]Microsoft Research, Redmond [3]Duke University
{xwang,yfwang,william}@cs.ucsb.edu
{qihua,aslicel,jfgao,leizhang}.microsoft.com, dinghan.shen@duke.edu

Abstract

Vision-language navigation (VLN) is the task of navigating an embodied agent to carry out natural language instructions inside real 3D environments. In this paper, we study how to address three critical challenges for this task: the cross-modal grounding, the ill-posed feedback, and the generalization problems. First, we propose a novel Reinforced Cross-Modal Matching (RCM) approach that enforces cross-modal grounding both locally and globally via reinforcement learning (RL). Particularly, a matching critic is used to provide an intrinsic reward to encourage global matching between instructions and trajectories, and a reasoning navigator is employed to perform cross-modal grounding in the local visual scene. Evaluation on a VLN benchmark dataset shows that our RCM model significantly outperforms existing methods by 10% on SPL and achieves the new state-of-the-art performance. To improve the generalizability of the learned policy, we further introduce a Self-Supervised Imitation Learning (SIL) method to explore unseen environments by imitating its own past, good decisions. We demonstrate that SIL can approximate a better and more efficient policy, which tremendously minimizes the success rate performance gap between seen and unseen environments (from 30.7% to 11.7%).

1. Introduction

Recently, vision-language grounded embodied agents have received increased attention [32, 22, 7] due to their popularity in many intriguing real-world applications, *e.g.*, in-home robots and personal assistants. Meanwhile, such an agent pushes forward visual and language grounding by putting itself in an active learning scenario through first-person vision. In particular, Vision-Language Navigation (VLN) [2] is the task of navigating an agent inside real environments by following natural language instructions. VLN

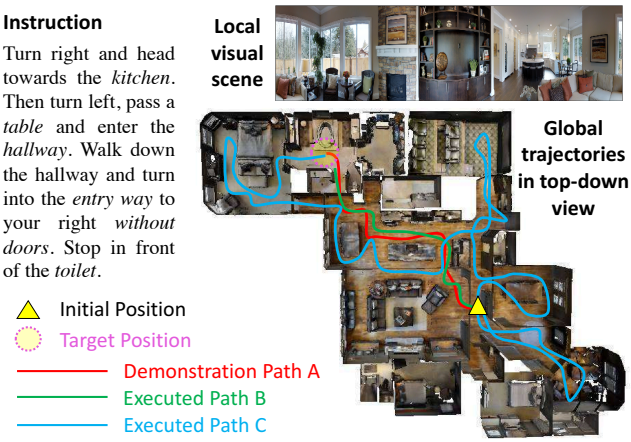


Figure 1: Demonstration of the VLN task. The instruction, the local visual scene, and the global trajectories in a top-down view is shown. The agent does not have access to the top-down view. Path A is the demonstration path following the instruction. Path B and C are two different paths executed by the agent.

requires a deep understanding of linguistic semantics, visual perception, and most importantly, the alignment of the two. The agent must reason about the vision-language dynamics in order to move towards the target that is inferred from the instructions.

VLN presents some unique challenges. First, reasoning over visual images and natural language instructions can be difficult. As we demonstrate in Figure 1, to reach a destination, the agent needs to ground an instruction in the local visual scene, represented as a sequence of words, as well as match the instruction to the visual trajectory in the global temporal space. Secondly, except for strictly following expert demonstrations, the feedback is rather coarse, since the "Success" feedback is provided only when the agent reaches a target position, completely ignoring whether the agent has followed the instructions (*e.g.*, Path A in Figure 1)

or followed a random path to reach the destination (*e.g.*, Path C in Figure 1). Even a "good" trajectory that matches an instruction can be considered unsuccessful if the agent stops marginally earlier than it should be (*e.g.*, Path B in Figure 1). An ill-posed feedback can potentially deviate from the optimal policy learning. Thirdly, existing work suffers from the generalization problem, causing a huge performance gap between seen and unseen environments.

In this paper, we propose to combine the power of reinforcement learning (RL) and imitation learning (IL) to address the challenges above. First, we introduce a novel Reinforced Cross-Modal Matching (RCM) approach that enforces cross-modal grounding both locally and globally via RL. Specifically, we design a *reasoning navigator* that learns the cross-modal grounding in both the textual instruction and the local visual scene, so that the agent can infer which sub-instruction to focus on and where to look at. From the global perspective, we equip the agent with a *matching critic* that evaluates an executed path by the probability of reconstructing the original instruction from it, which we refer to as the cycle-reconstruction reward. Locally, the cycle-reconstruction reward provides a fine-grained intrinsic reward signal to encourage the agent to better understand the language input and penalize the trajectories that do not match the instructions. For instance, using the proposed reward, Path B is considered better than Path C (see Figure 1).

Being trained with the intrinsic reward from the matching critic and the extrinsic reward from the environment, the reasoning navigator learns to ground the natural language instruction on both local spatial visual scene and global temporal visual trajectory. Our RCM model significantly outperforms the existing methods and achieves new state-of-the-art performance on the Room-to-Room (R2R) dataset.

Our experimental results indicate a large performance gap between seen and unseen environments. To narrow the gap, we propose an effective solution to explore unseen environments with self-supervision. This technique is valuable because it can facilitate lifelong learning and adaption to new environments. For example, in-home robots can explore a new home it arrives at and iteratively improve the navigation policy by learning from previous experience. Motivated by this fact, we introduce a Self-Supervised Imitation Learning (SIL) method in favor of exploration on unseen environments that do not have labeled data. The agent learns to imitate its own past, good experience. Specifically, in our framework, the navigator performs multiple roll-outs, of which good trajectories (determined by the matching critic) are stored in the replay buffer and later used for the navigator to imitate. In this way, the navigator can approximate its best behavior that leads to a better policy. To summarize, our contributions are mainly four-fold:

- We propose a novel Reinforced Cross-Modal Matching (RCM) framework that utilizes both extrinsic and intrinsic rewards for reinforcement learning, of which we introduce a cycle-reconstruction reward as the intrinsic reward to enforce the global matching between the language instruction and the agent's trajectory.
- Our reasoning navigator learns the cross-modal contexts and makes decisions based on trajectory history, textual context, and visual context.
- Experiments show that RCM achieves the new state-of-the-art performance on the R2R dataset, and among the prior art, is ranked first in the VLN Challenge in terms of SPL, the most reliable metric for the task.
- In addition, we introduce a Self-Supervised Imitation Learning (SIL) method to explore the unseen environments with self-supervision, and validate its effectiveness and efficiency on the R2R dataset.

2. Related Work

Vision-and-Language Grounding Recently, researchers in both computer vision and natural language processing are striving to bridge vision and natural language towards a deeper understanding of the world [46, 40, 20, 6, 24, 17, 37, 19], *e.g.*, captioning an image or a video with natural language [9, 10, 39, 41, 47, 48, 42] or localizing desired objects within an image given a natural language description [31, 18, 49]. Moreover, visual question answering [3] and visual dialog [8] aim to generate one-turn or multi-turn response by grounding it on both visual and textual modalities. However, those tasks focus on passive visual perception in the sense that the visual inputs are usually fixed. In this work, we are particularly interested in solving the dynamic multi-modal grounding problem in both temporal and spatial spaces. Thus, we focus on the task of vision-language navigation (VLN) [2] which requires the agent to actively interact with the environment.

Embodied Navigation Agent Navigation in 3D environments [50, 25, 26, 14] is an essential capability of a mobile intelligent system that functions in the physical world. In the past two years, a plethora of tasks and evaluation protocols [32, 22, 34, 45, 2] have been proposed as summarized in [1]. VLN [2] focuses on language-grounded navigation in the real 3D environment. In order to solve the VLN task, Anderson *et al.* [2] set up an attention-based sequence-to-sequence baseline model. Then Wang *et al.* [43] introduced a hybrid approach that combines model-free and model-based reinforcement learning (RL) to improve the model's generalizability. Lately, Fried *et al.* [11] proposed a speaker-follower model that adopts data augmentation, panoramic action space and modified beam search for VLN, establishing the current state-of-the-art performance on the

Room-to-Room dataset. Extending prior work, we propose a Reinforced Cross-Modal Matching (RCM) approach to VLN. The RCM model is built upon [11] but differs in many significant aspects: (1) we combine a novel multi-reward RL with imitation learning for VLN while Speaker-Follower models [11] only uses supervised learning as in [2]. (2) Our reasoning navigator performs cross-modal grounding rather than the temporal attention mechanism on single-modality input. (3) Our matching critic is similar to Speaker in terms of the architecture design, but the former is used to provide the cycle-reconstruction intrinsic reward for both RL and SIL training while the latter is used to augment training data for supervised learning. Moreover, we introduce a self-supervised imitation learning method for exploration in order to explicitly address the generalization issue, which is a problem not well-studied in prior work.

Exploration Much work has been done on improving exploration [4, 12, 16, 28, 36] because the trade-off between exploration and exploitation is one of the fundamental challenges in RL. The agent needs to exploit what it has learned to maximize reward and explore new territories for better policy search. Curiosity or uncertainty has been used as a signal for exploration [33, 35, 23, 29]. Most recently, Oh *et al*. [27] proposed to exploit past good experience for better exploration in RL and theoretically justified its effectiveness. Our Self-Supervised Imitation Learning (SIL) method shares the same spirit. But instead of testing on games, we, for the first time, adapt SIL and validate its effectiveness and efficiency on the more practical task of VLN.

3. Reinforced Cross-Modal Matching

3.1. Overview

Here we consider an embodied agent that learns to navigate inside real indoor environments by following natural language instructions. As described in Figure 2, the RCM framework mainly consists of two modules: a **reasoning navigator** π_θ and a **matching critic** V_β. Given the initial state s_0 and the natural language instruction (a sequence of words) $\mathcal{X} = x_1, x_2, ..., x_n$, the reasoning navigator learns to perform a sequence of actions $a_1, a_2, ..., a_T \in \mathcal{A}$, which generates a trajectory τ, in order to arrive at the target location s_{target} indicated by the instruction $\mathcal{X}$. The navigator interacts with the environment and perceives new visual states as it executes actions. To promote the generalizability and reinforce the policy learning, we introduce two reward functions: an **extrinsic reward** that is provided by the environment and measures the success signal and the navigation error of each action, and an **intrinsic reward** that comes from our matching critic and measures the alignment between the language instruction $\mathcal{X}$ and the navigator's trajectory τ.

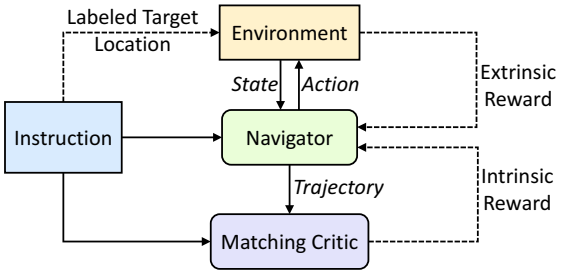


Figure 2: Overview of our RCM framework.

3.2. Model

Here we discuss the reasoning navigator and matching critic in details, both of which are end-to-end trainable.

3.2.1 Cross-Modal Reasoning Navigator

The navigator π_θ is a policy-based agent that maps the input instruction $\mathcal{X}$ onto a sequence of actions $\{a_t\}_{t=1}^T$. At each time step t, the navigator receives a state s_t (the visual scene) from the environment and needs to ground the textual instruction in the local visual scene. Thus, we design a cross-modal reasoning navigator that learns the trajectory history, the focus of the textual instruction, and the local visual attention in order, which forms a cross-modal reasoning path to encourage the local dynamics of both modalities at step t.

Figure 3 shows the unrolled version of the navigator at time step t. Similar to [11], we equip the navigator with a panoramic view, which is split into image patches of m different viewpoints, so the panoramic features that are extracted from the visual state s_t can be represented as $\{v_{t,j}\}_{j=1}^m$, where $v_{t,j}$ denotes the pre-trained CNN feature of the image patch at viewpoint j.

History Context Once the navigator runs one step, the visual scene would change accordingly. The history of the trajectory $\tau_{1:t}$ till step t is encoded as a history context vector h_t by an attention-based trajectory encoder LSTM [15]:

$$h_t = LSTM([v_t, a_{t-1}], h_{t-1}) \quad (1)$$

where a_{t-1} is the action taken at previous step, and $v_t = \sum_j \alpha_{t,j} v_{t,j}$, the weighted sum of the panoramic features. $\alpha_{t,j}$ is the attention weight of the visual feature $v_{t,j}$, representing its importance with respect to the previous history context h_{t-1}. Note that we adopt the dot-product attention [38] hereafter, which we denote as (taking the attention over visual features above for an example)

$$v_t = attention(h_{t-1}, \{v_{t,j}\}_{j=1}^m) \quad (2)$$

$$= \sum_j softmax(h_{t-1} W_h (v_{t,j} W_v)^T) v_{t,j} \quad (3)$$

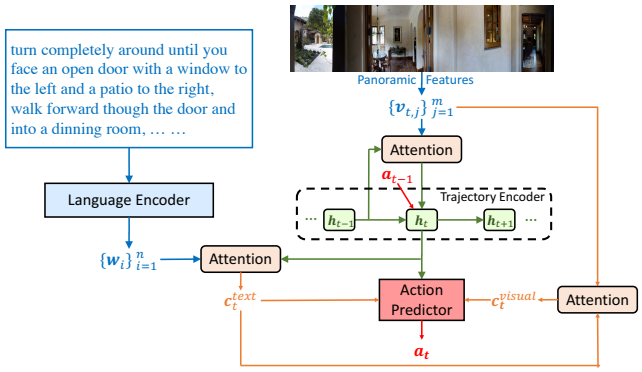


Figure 3: Cross-modal reasoning navigator at step t.

where W_h and W_v are learnable projection matrices.

Visually Conditioned Textual Context Memorizing the past can enable the recognition of the current status and thus understanding which words or sub-instructions to focus on next. Hence, we further learn the textual context c_t^{text} conditioned on the history context h_t. We let a language encoder LSTM to encode the language instruction $\mathcal{X}$ into a set of textual features $\{w_i\}_{i=1}^n$. Then at every time step, the textual context is computed as

$$c_t^{text} = attention(h_t, \{w_i\}_{i=1}^n) \tag{4}$$

Note that c_t^{text} weighs more on the words that are more relevant to the trajectory history and the current visual state.

Textually Conditioned Visual Context Knowing where to look at requires a dynamic understanding of the language instruction; so we compute the visual context c_t^{visual} based on the textual context c_t^{text}:

$$c_t^{visual} = attention(c_t^{text}, \{v_j\}_{j=1}^m) \tag{5}$$

Action Prediction In the end, our action predictor considers the history context h_t, the textual context c_t^{text}, and the visual context c_t^{visual}, and decides which direction to go next based on them. It calculates the probability p_k of each navigable direction using a bilinear dot product as follows:

$$p_k = softmax([h_t, c_t^{text}, c_t^{visual}]W_c(u_k W_u)^T) \tag{6}$$

where u_k is the action embedding that represents the k-th navigable direction, which is obtained by concatenating an appearance feature vector (CNN feature vector extracted from the image patch around that view angle or direction) and a 4-dimensional orientation feature vector $[sin\psi; cos\psi; sin\omega; cos\omega]$, where ψ and ω are the heading and elevation angles respectively. The learning objectives for training the navigator are introduced in Section 3.3.

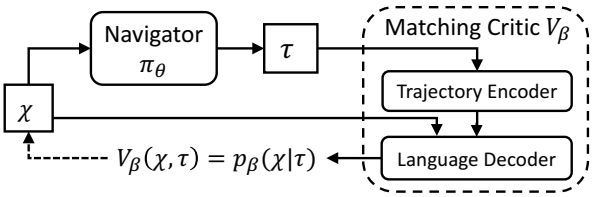


Figure 4: Cross-modal matching critic that provides the cycle-reconstruction intrinsic reward.

3.2.2 Cross-Modal Matching Critic

In addition to the extrinsic reward signal from the environment, we also derive an intrinsic reward R_{intr} provided by the matching critic V_β to encourage the global matching between the language instruction $\mathcal{X}$ and the navigator π_θ's trajectory $\tau = \{< s_1, a_1 >, < s_2, a_2 >, ..., < s_T, a_T >\}$:

$$R_{intr} = V_\beta(\mathcal{X}, \tau) = V_\beta(\mathcal{X}, \pi_\theta(\mathcal{X})) \tag{7}$$

One way to realize this goal is to measure the cycle-reconstruction reward $p(\hat{\mathcal{X}} = \mathcal{X}|\pi_\theta(\mathcal{X}))$, the probability of reconstructing the language instruction $\mathcal{X}$ given the trajectory $\tau = \pi_\theta(\mathcal{X})$ executed by the navigator. The higher the probability is, the better the produced trajectory is aligned with the instruction.

Therefore as shown in Figure 4, we adopt an attention-based sequence-to-sequence language model as our matching critic V_β, which encodes the trajectory τ with a trajectory encoder and produces the probability distributions of generating each word of the instruction $\mathcal{X}$ with a language decoder. Hence the intrinsic reward

$$R_{intr} = p_\beta(\mathcal{X}|\pi_\theta(\mathcal{X})) = p_\beta(\mathcal{X}|\tau) \tag{8}$$

which is normalized by the instruction length n. In our experiments, the matching critic is pre-trained with human demonstrations (the ground-truth instruction-trajectory pairs $< \mathcal{X}^*, \tau^* >$) via supervised learning.

3.3. Learning

In order to quickly approximate a relatively good policy, we use the demonstration actions to conduct supervised learning with maximum likelihood estimation (MLE). The training loss L_{sl} is defined as

$$L_{sl} = -\mathbb{E}[\log(\pi_\theta(a_t^*|s_t))] \tag{9}$$

where a_t^* is the demonstration action provided by the simulator. Warm starting the agent with supervised learning can ensure a relatively good policy on the seen environments. But it also limits the agent's generalizability to recover from erroneous actions in unseen environments, since it only clones the behaviors of expert demonstrations.

To learn a better and more generalizable policy, we then switch to reinforcement learning and introduce the extrinsic and intrinsic reward functions to refine the policy from different perspectives.

Extrinsic Reward A common practice in RL is to directly optimize the evaluation metrics. Since the objective of the VLN task is to successfully reach the target location s_{target}, we consider two metrics for the reward design. The first metric is the relative navigation distance similar to [43]. We denote the distance between s_t and s_{target} as $\mathcal{D}_{target}(s_t)$. Then the immediate reward $r(s_t, a_t)$ after taking action a_t at state s_t ($t < T$) becomes:

$$r(s_t, a_t) = \mathcal{D}_{target}(s_t) - \mathcal{D}_{target}(s_{t+1}), \quad t < T \tag{10}$$

This indicates the reduced distance to the target location after taking action a_t. Our second choice considers the "Success" as an additional criterion. If the agent reaches a point within a threshold measured by the distance d from the target (d is preset as 3m in the R2R dataset), then it is counted as "Success". Particularly, the immediate reward function at last step T is defined as

$$r(s_T, a_T) = \mathbb{1}(\mathcal{D}_{target}(s_T) \leq d) \tag{11}$$

where $\mathbb{1}()$ is an indicator function. To incorporate the influence of the action a_t on the future and account for the local greedy search, we use the discounted cumulative reward rather than the immediate reward to train the policy:

$$R_{extr}(s_t, a_t) = \underbrace{r(s_t, a_t)}_{\text{immediate reward}} + \underbrace{\sum_{t'=t+1}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})}_{\text{discounted future reward}} \tag{12}$$

where γ is the discounted factor (0.95 in our experiments).

Intrinsic Reward As discussed in Section 3.2.2, we pretrain a matching critic to calculate the cycle-reconstruction intrinsic reward R_{intr} (see Equation 8), promoting the alignment between the language instruction $\mathcal{X}$ and the trajectory τ. It encourages the agent to respect the instruction and penalizes the paths that deviate from what the instruction indicates.

With both the extrinsic and intrinsic reward functions, the RL loss can be written as

$$L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta}[A_t] \tag{13}$$

where the advantage function $A_t = R_{extr} + \delta R_{intr} - b_t$. δ is a hyperparameter weighing the intrinsic reward, and b_t is the estimated baseline to reduce the variance, which is a linear regressor with the trajectory encoder's hidden state h_t as the input. Based on REINFORCE algorithm [44], the gradient of non-differentiable, reward-based loss function can be derived as

$$\nabla_\theta L_{rl} = -A_t \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{14}$$

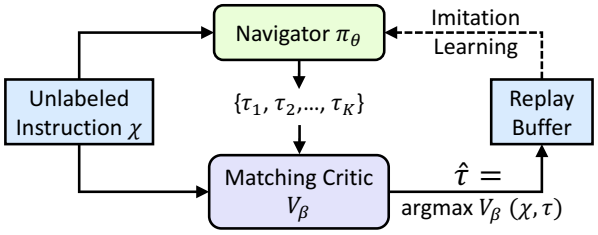


Figure 5: SIL for exploration on unlabeled data.

4. Self-Supervised Imitation Learning

The last section introduces the effective RCM method for generic vision-language navigation task, whose standard setting is to train the agent on seen environments and test it on unseen environments without exploration. In this section we discuss a different setting where the agent is allowed to explore unseen environments without ground-truth demonstrations. This is of practical benefit because it facilitates lifelong learning and adaption to new environments.

To this end, we propose a Self-Supervised Imitation Learning (SIL) method to imitate the agent's own past good decisions. As shown in Figure 5, given a natural language instruction $\mathcal{X}$ without paired demonstrations and ground-truth target location, the navigator produces a set of possible trajectories and then stores the best trajectory $\hat{\tau}$ that is determined by matching critic V_β into a replay buffer, in formula,

$$\hat{\tau} = \arg\max_{\tau} V_\beta(\mathcal{X}, \tau) \tag{15}$$

The matching critic evaluates the trajectories with the cycle-reconstruction reward as introduced in Section 3.2.2. Then by exploiting the good trajectories in the replay buffer, the agent is indeed optimizing the following objective with self-supervision. The target location is unknown and thus there is no supervision from the environment.

$$L_{sil} = -R_{intr} \log \pi_\theta(a_t|s_t) \tag{16}$$

Note that L_{sil} can be viewed as the loss for policy gradient except that the off-policy Monte-Carlo return R_{intr} is used instead of on-policy return. L_{sil} can also be interpreted as the supervised learning loss with $\hat{\tau}$ as the "ground truths":

$$L_{sil} = -\mathbb{E}[\log(\pi_\theta(\hat{a}_t|s_t))] \tag{17}$$

where $\hat{a}_t$ is the action stored in the replay buffer using Equation 15. Paired with a matching critic, the SIL method can be combined with various learning methods to approximate a better policy by imitating the previous best of itself.

5. Experiments and Analysis

5.1. Experimental Setup

R2R Dataset We evaluate our approaches on the Room-to-Room (R2R) dataset [2] for vision-language navigation in real 3D environments, which is built upon the Matterport3D dataset [5]. The R2R dataset has 7,189 paths that capture most of the visual diversity and 21,567 human-annotated instructions with an average length of 29 words. The R2R dataset is split into training, seen validation, unseen validation, and test sets. The seen validation set shares the same environments with the training set. While both the unseen validation and test sets contain distinct environments that do not appear in the other sets.

Testing Scenarios The standard testing scenario of the VLN task is to train the agent in seen environments and then test it in previously unseen environments in a zero-shot fashion. There is no prior exploration on the test set. This setting is preferred and able to clearly measure the generalizability of the navigation policy, so we evaluate our RCM approach under the standard testing scenario.

Furthermore, exploration in unseen environments is certainly meaningful in practice, *e.g.*, in-home robots are expected to explore and adapt to a new environment. So we introduce a lifelong learning scenario where the agent is encouraged to learn from trials and errors on the unseen environments. In this case, how to effectively explore the unseen validation or test set where there are no expert demonstrations becomes an important task to study.

Evaluation Metrics We report five evaluation metrics as used by the VLN Challenge: Path Length (PL), Navigation Error (NE), Oracle Success Rate (OSR), Success Rate (SR), and Success rate weighted by inverse Path Length (SPL).[1] Among those metrics, SPL is the recommended primary measure of navigation performance [1], as it considers both effectiveness and efficiency. The other metrics are also reported as auxiliary measures.

Implementation Details Following prior work [2, 43, 11], ResNet-152 CNN features [13] are extracted for all images without fine-tuning. The pretrained GloVe word embeddings [30] are used for initialization and then fine-tuned during training. We train the matching critic with human demonstrations and then fix it during policy learning. Then we warm start the policy via SL with a learning rate 1e-4, and then switch to RL training with a learning rate 1e-5 (same for SIL). Adam optimizer [21] is used to optimize all the parameters. More details can be found in the appendix.

[1]PL: the total length of the executed path. NE: the shortest-path distance between the agent's final position and the target. OSR: the success rate at the closest point to the goal that the agent has visited along the trajectory. SR: the percentage of predicted end-locations within 3m of the target locations. SPL: SPL trades-off Success Rate against Path Length, which is defined in [1].

Test Set (VLN Challenge Leaderboard)					
Model	PL ↓	NE ↓	OSR ↑	SR ↑	**SPL ↑**
Random	9.89	9.79	18.3	13.2	12
seq2seq [2]	**8.13**	7.85	26.6	20.4	18
RPA [43]	9.15	7.53	32.5	25.3	23
Speaker-Follower [11]	14.82	6.62	44.0	35.0	28
+ beam search	*1257.38*	4.87	96.0	53.5	*1*
Ours					
RCM	15.22	**6.01**	**50.8**	**43.1**	35
RCM + SIL (train)	**11.97**	6.12	49.5	43.0	**38**
RCM + SIL (unseen)[2]	9.48	4.21	66.8	60.5	59

Table 1: Comparison on the R2R test set [2]. Our RCM model significantly outperforms the SOTA methods, especially on SPL (the primary metric for navigation tasks [1]). Moreover, using SIL to imitate itself on the training set can further improve its efficiency: the path length is shortened by 3.25m. Note that with beam search, the agent executes K trajectories at test time and chooses the most confident one as the ending point, which results in a super long path and is heavily penalized by SPL.

5.2. Results on the Test Set

Comparison with SOTA We compare the performance of RCM to the previous state-of-the-art (SOTA) methods on the test set of the R2R dataset, which is held out as the VLN Challenge. The results are shown in Table 1, where we compare RCM to a set of baselines: (1) *Random*: randomly take a direction to move forward at each step until five steps. (2) *seq2seq*: the best-performing sequence-to-sequence model as reported in the original dataset paper [2], which is trained with the student-forcing method. (3) *RPA*: a reinforced planning-ahead model that combines model-free and model-based reinforcement learning for VLN [43]. (4) *Speaker-Follower*: a compositional Speaker-Follower method that combines data augmentation, panoramic action space, and beam search for VLN [11].

As can be seen in Table 1, RCM significantly outperforms the existing methods, improving the SPL score from 28% to 35%[3]. The improvement is consistently observed on the other metrics, *e.g.*, the success rate is increased by 8.1%. Moreover, using SIL to imitate the RCM agent's previous best behaviors on the training set can approximate a more efficient policy, whose average path length is reduced

[2]The results of using SIL to explore unseen environments are only used to validate its effectiveness for lifelong learning, which is not directly comparable to other models due to different learning scenarios.

[3]Note that our RCM model also utilizes the panoramic action space and augmented data in [11] for a fair comparison.

#	Model	Seen Validation				Unseen Validation			
		PL ↓	NE ↓	OSR ↑	SR ↑	PL ↓	NE ↓	OSR ↑	SR ↑
0	Speaker-Follower (no beam search) [11]	-	3.36	73.8	66.4	-	6.62	45.0	35.5
1	RCM + SIL (train)	**10.65**	3.53	75.0	66.7	**11.46**	6.09	50.1	**42.8**
2	RCM	11.92	3.37	76.6	67.4	14.84	**5.88**	**51.9**	42.5
3	– intrinsic reward	12.08	3.25	**77.2**	**67.6**	15.00	6.02	50.5	40.6
4	– extrinsic reward = pure SL	11.99	3.22	76.7	66.9	14.83	6.29	46.5	37.7
5	– cross-modal reasoning	11.88	**3.18**	73.9	66.4	14.51	6.47	44.8	35.7
6	RCM + SIL (unseen)	**10.13**	**2.78**	**79.7**	**73.0**	**9.12**	**4.17**	**69.31**	**61.3**

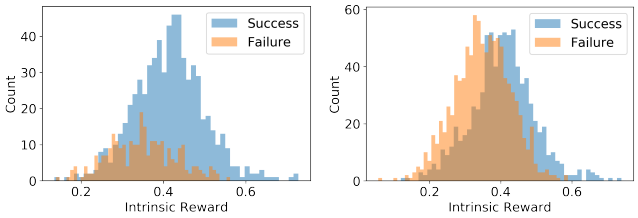
Table 2: Ablation study on seen and unseen validation sets. We report the performance of the speaker-follower model without beam search as the baseline. Row 1-5 shows the influence of each individual component by successively removing it from the final model. Row 6 illustrates the power of SIL on exploring unseen environments with self-supervision. Please see Section 5.3 for more detailed analysis.

from 15.22m to 11.97m and which achieves the best result (38%) on SPL. Therefore, we submit the results of *RCM + SIL (train)* to the VLN Challenge, ranking first among prior work in terms of SPL. It is worth noticing that beam search is not practical in reality, because it needs to execute a very long trajectory before making the decision, which is punished heavily by the primary metric SPL. So we are mainly comparing the results without beam search.

Self-Supervised Imitation Learning As mentioned above, for a standard VLN setting, we employ SIL on the training set to learn an efficient policy. For the lifelong learning scenario, we test the effectiveness of SIL on exploring unseen environments (the validation and test sets). It is noticeable in Table 1 that SIL indeed leads to a better policy even without knowing the target locations. SIL improves RCM by 17.5% on SR and 21% on SPL. Similarly, the agent also learns a more efficient policy that takes less number of steps (the average path length is reduced from 15.22m to 9.48m) but obtains a higher success rate. The key difference between SIL and beam search is that SIL optimizes the policy itself by play-and-imitate while beam search only makes a greedy selection of the rollouts of the existing policy. But we would like to point out that due to different learning scenarios, the results of *RCM + SIL (unseen)* cannot be directly compared with other methods following the standard settings of the VLN challenge.

5.3. Ablation Study

Effect of Individual Components We conduct an ablation study to illustrate the effect of each component on both seen and unseen validation sets in Table 2. Comparing Row 1 and Row 2, we observe the efficiency of the learned policy by imitating the best of itself on the training set. Then we start with the RCM model in Row 2, and successively remove the *intrinsic reward*, *extrinsic reward*, and *cross-modal reasoning* to demonstrate their importance.



(a) Seen Validation (b) Unseen Validation

Figure 6: Visualization of the intrinsic reward on seen and unseen validation sets.

Removing the intrinsic reward (Row 3), we notice that the success rate (SR) on unseen environments drops 1.9 points while it is almost fixed on seen environments (0.2↑). It evaluates the alignment between instructions and trajectories, serving as a complementary supervision besides of the feedback from the environment, therefore it works better for the unseen environments that require more supervision due to lack of exploration. This also indirectly validates the importance of exploration on unseen environments.

Furthermore, the results of Row 4 (the RCM model with only supervised learning) validate the superiority of reinforcement learning compared to purely supervised learning on the VLN task. Meanwhile, since eventually the results are evaluated based on the success rate (SR) and path length (PL), directly optimizing the extrinsic reward signals can guarantee the stability of the reinforcement learning and bring a big performance gain.

We then verify the strength of our cross-modal reasoning navigator by comparing it (Row 4) with an attention-based sequence-to-sequence model (Row 5) that utilizes the previous hidden state h_{t-1} to attend to both the visual and textual features at decoding time. Everything else is exactly the same except the cross-modal attention design. Evidently, our navigator improves upon the baseline by considering

(a) A successful case (b) A failure case

Figure 7: Qualitative examples from the unseen validation set.

history context, visually-conditioned textual context, and textually-conditioned visual context for decision making.

In the end, we demonstrate the effectiveness of the proposed SIL method for exploration in Row 6. Considerable performance boosts have been obtained on both seen and unseen environments, as the agent learns how to better execute the instructions from its own previous experience.

Generalizability Another observation from the experiments (*e.g.*, see Table 2) is that our RCM approach is much more generalizable to unseen environments compared with the baseline. The improvements on the seen and unseen validation sets are 0.3 and 7.1 points, respectively. So is the SIL method, which explicitly explores the unseen environments and tremendously reduces the success rate performance gap between seen and unseen environments from 30.7% (Row 5) to 11.7% (Row 6).

Visualizing Intrinsic Reward In Figure 6, we plot the histogram distributions of the intrinsic rewards (produced by our submitted model) on both seen and unseen validation sets. On the one hand, the intrinsic reward is aligned with the success rate to some extent, because the successful examples are receiving higher averaged intrinsic rewards than the failed ones. On the other hand, the complementary intrinsic reward provides more fine-grained reward signals to reinforce multi-modal grounding and improve the navigation policy learning.

Qualitative Analysis For a more intuitive view of how our model works for the VLN task, we visualize two qualitative examples in Figure 7. Particularly, we choose two examples, both with high intrinsic rewards. In (a), the agent successfully reaches the target destination, with a comprehensive understanding of the natural language instruction. While in (b), the intrinsic reward is also high, which indicates most of the agent's actions are good, but it is also noticeable that the agent fails to recognize *the laundry room* at the end of the trajectory, which shows the importance of more precise visual grounding in the navigation task.

6. Conclusion

In this paper we present two novel approaches, RCM and SIL, which combine the strength of reinforcement learning and self-supervised imitation learning for the vision-language navigation task. Experiments illustrate the effectiveness and efficiency of our methods under both the standard testing scenario and the lifelong learning scenario. Moreover, our methods show strong generalizability in unseen environments. Note that the proposed learning frameworks are modular and model-agnostic, which allow the components to be improved separately. We also believe that these methods can be easily generalized to other tasks.

Acknowledgment

This work was partly performed when the first author was interning at Microsoft Research. The authors thank Peter Anderson and Pengchuan Zhang for their helpful discussions, and Ronghang Hu for his visualization code.

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, 6

[2] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. 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)*, volume 2, 2018. 1, 2, 3, 6

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[4] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016. 3

[5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 6

[6] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431, 2015. 2

[7] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[8] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[10] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 2

[11] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2, 3, 6, 7

[12] J. Gao, M. Galley, and L. Li. Neural approaches to conversational ai. *arXiv preprint arXiv:1809.08267*, 2018. 3

[13] 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*, pages 770–778, 2016. 6

[14] S. Hemachandra, F. Duvallet, T. M. Howard, N. Roy, A. Stentz, and M. R. Walter. Learning models for following natural language directions in unknown environments. *arXiv preprint arXiv:1503.05079*, 2015. 2

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[16] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016. 3

[17] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016. 2

[18] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 2

[19] Q. Huang, P. Zhang, D. Wu, and L. Zhang. Turbo learning for captionbot and drawingbot. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2

[20] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] 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. 1, 2

[23] Z. C. Lipton, J. Gao, L. Li, X. Li, F. Ahmed, and L. Deng. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. *arXiv preprint arXiv:1608.05081*, 2016. 3

[24] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 2

[25] 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. 2

[26] A. Mousavian, A. Toshev, M. Fiser, J. Kosecka, and J. Davidson. Visual representations for semantic target driven navigation. *arXiv preprint arXiv:1805.06066*, 2018. 2

[27] J. Oh, Y. Guo, S. Singh, and H. Lee. Self-imitation learning. *arXiv preprint arXiv:1806.05635*, 2018. 3

[28] G. Ostrovski, M. G. Bellemare, A. v. d. Oord, and R. Munos. Count-based exploration with neural density models. *arXiv preprint arXiv:1703.01310*, 2017. 3

[29] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017. 3

[30] 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)*, pages 1532–1543, 2014. 6

[31] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 2

[32] 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. 1, 2

[33] J. Schmidhuber. Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer, 1991. 3

[34] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[35] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008. 3

[36] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2753–2762, 2017. 3

[37] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. 2

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 3

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015. 2

[40] X. Wang, W. Chen, Y.-F. Wang, and W. Y. Wang. No metrics are perfect: Adversarial reward learning for visual storytelling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. 2

[41] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[42] X. Wang, Y.-F. Wang, and W. Y. Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018. 2

[43] X. Wang, W. Xiong, H. Wang, and W. Y. Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 5, 6

[44] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 5

[45] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 2

[46] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 2

[47] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016. 2

[48] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016. 2

[49] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 2

[50] 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 *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017. 2