Landmark-RxR: Solving Vision-and-Language Navigation with Fine-Grained Alignment Supervision

Anonymous Author(s)
Affiliation
Address
email

Abstract

In Vision-and-Language Navigation (VLN) task, an agent is asked to navigate inside 3D indoor environments following given instructions. Cross-modal alignment is one of the most critical challenges in VLN because the prediction trajectory needs to match the given instruction accurately. In this paper, we address the cross-modal alignment challenge from a fine-grained perspective. Firstly, to alleviate weak cross-modal alignment supervision from coarse-grained data, we introduce a human-annotated fine-grained VLN dataset, namely Landmark-RxR. Secondly, to further enhance local cross-modal alignment under fine-grained supervision, we investigate the focal-oriented rewards with soft and hard formations, by focusing on the critical points sampled from fine-grained Landmark-RxR. Moreover, to fully evaluate the navigation process, we also propose a re-initialization mechanism that makes metrics insensitive to difficult points, which can cause the agent to deviate from the correct trajectories. Experimental results show that our agent has superior navigation performance on Landmark-RxR, en-RxR and R2R.

1 Introduction

Vision-and-language navigation (VLN) is an important task about cross-modal intelligence and can be applied to service and rescue robots. Different from other cross-modal tasks like image/video captioning [1, 2] and visual question answering [3], where the agent only needs to understand fixed images or videos, VLN agent has to learn and reason by dynamically interacting with the real environment guided by human instructions. Since Anderson *et al.* [4] first introduced the VLN task with the coarse-grained dataset Room-to-Room (R2R), great progresses in this direction have been made, ranging from the sequence-to-sequence models [4] combined with cross-modal grounding modules [5–8] to Transformer-based models [7, 9–12].

Cross-modal alignment is one of the most critical challenges in VLN as the prediction trajectory needs to match the given instruction accurately. However, it is difficult for the agent to learn very accurate visual and textual modality alignment from coarse-grained data, *i.e.*, only coarse-level correspondences between global instructions and trajectories are annotated, without finer ones between sub-instructions and sub-trajectories. Aforementioned works [4–8, 10–12] tried attending on specific parts of the instructions during navigation to help the agent better align salient visual contents with the textual modality. But only coarse-grained data like R2R are not capable of providing enough supervising signals of accurate cross-modal alignment. Some recent works such as Fine-Grained R2R [13] and BabyWalk [14] also found this same issue and proposed fine-grained datasets based on R2R. However, their datasets are generated by heuristic rules and are not precise enough, which inevitably limited the navigation performance. Although Room-across-Room (RxR) dataset [15] includes word-level alignment between each word and point in the trajectory, it lacks segmentation tags to split an instruction into sub-instructions and landmarks, namely the marked end points of sub-trajectories. Since words in the same sub-instruction are highly correlated, these important correlations are lost

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

because no sub-instruction can we get from RxR. It is also the same for the sub-trajectories. As a result, there is also an absence of the correlation between the words in a sub-instruction and the points in corresponding sub-trajectories, which should be an important cross-modal alignment supervision signal.

In addition to the dataset annotation, how to design better rewards during reinforcement learning is another important issue on cross-modal alignment learning. Currently, reward shaping has been well investigated in VLN based on coarse-grained data to align instructions and trajectories globally, but pays little attention to enhance local cross-modal alignment. The goal-oriented reward simply [6] uses the arrival signal and reduced distance to the goal point as reward, guiding agent to find the global goal points. The fidelity-oriented rewards [16, 17] help agent improve the similarity between the prediction trajectory and demonstration trajectory globally by taking intermediate points as external supervision. However, for points in trajectory, some critical ones are more helpful to navigation process, *i.e.*, the intermediate points with more detailed description in the instruction can assist agent in correcting its trajectory in time and reaching the locations that the instruction really concern. Thus, an emphasis on alignment between words in critical parts of the instructions and corresponding critical points in trajectories, which is in an local to local way, is essential. And our Landmark-RxR could just provide the required fine-grained supervision for better local cross-modal alignment learning.

In this paper, we address the challenges above from a fine-grained perspective. Firstly, based on english Guide part of RxR, we introduce a fine-grained dataset Landmark-RxR, which is human-annotated, landmark-based, fine-grained and currently the largest scale. With the groundtruth fine-grained data, experiments demonstrate that agent generalizes better to unseen environment and instructions with domain gap. This indicates that the fine-grained data help agent to align textual and visual modality better. Secondly, we propose two kinds of focal-oriented rewards that encourage local alignment between the instructions and the critical points. For the critical points, we just sampled them from landmark set in our Landmark-RxR. The focal-oriented rewards outperform the commonly used goal-oriented reward and fidelity-oriented reward obviously. We also propose the re-initialization mechanism to fully evaluate the navigation process in a way that is insensitive to difficult points, which can cause the deviation from the correct trajectory. With the fine-grained data and focal-oriented rewards, our agent shows superior navigation performance on Landmark-RxR, en-RxR and R2R.

2 Related Work

Vision-and-language navigation Anderson *et al.* [4] first introduced the simulated, photo-realistic VLN task with a benchmark Room-to-Room (R2R) and several sequence-to-sequence baselines. Fried *et al.* [5] utilized the speaker-driven data and panoramic action space to improve navigation performance. Wang *et al.* [18] proposed a planned-ahead hybrid reinforcement learning model with model-free and model-based reinforcement learning for better generalization. Tan *et al.* [19] trained agent with mixed imitation learning and reinforcement learning firstly and used unseen triplets generated with environment dropout method for fine-tuning then. Jain *et al.* [16] and Ilharco *et al.* [17] proposed the fidelity-oriented rewards for reinforcement learning phrase to ensure the fidelity of prediction trajectories and given instructions. Kurita *et al.* [20] combined generative and discriminative policy to enhance the navigation ability from different aspect. Li *et al.* [7] leveraged the large-scale pretrained language models BERT [21] and GPT [22] for powerful text representation. Hao *et al.* [10] pretrained and finetuned generic transformer-based model in VLN environment and the pretrained model generalizes well to other two navigation tasks. Majumdar *et al.* [11] demonstrated that pretrained on image-text pairs from the web, the transformer-based model can perform on VLN task well after fine-tuning. Hong *et al.* [12] proposed a recurrent vision-and-language BERT to navigate in recurrent mechanism like LSTM efficiently.

Fine-grained VLN Wang *et al.* [6] designed a cross-modal grounding module for textual and visual representation, and refined navigation agent using reinforcement learning with goal-oriented reward. Ma *et al.* [8] introduced a self-monitoring agent with a visual-textual co-grounding module to locate the required part in instructions and a progress monitor to reflect navigation progress. Zhu *et al.* [14] adopted a two-stage method for navigation. The given instruction is firstly split into several sub-instructions named baby-steps according to heuristic rules and then the baby-steps are fed to the agent one by one. Similar to [14], Hong *et al.* [13] firstly segmented instruction into sub-instructions using a heuristic method and then the agent navigates the following sub-instructions sequentially

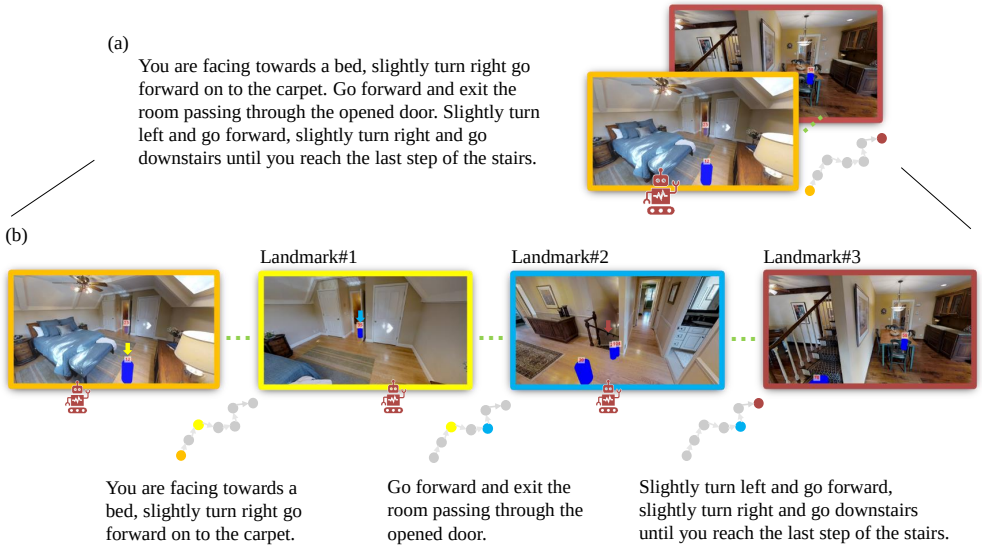

Figure 1: Three landmarks are marked in the complete trajectory and the corresponding sub-instructions are split out from the complete instruction. (a) An example of complete instruction and complete trajectory pair in en-RxR. (b) An example of three sub-instruction and sub-trajectory pairs split out from the complete instruction and complete trajectory pair. Each sub-instruction and sub-trajectory pair will be included in Landmark-RxR.

with a shifting module inferring whether the current sub-instruction has been completed. Ku *et al.* [15] explored utilizing spatially-temporally aligned priori to supervise the textual and visual attention weights but got a plain result. Different from these works, we focus on fine-grained supervision to benefit the cross-modal alignment. A human-annotated, fine-grained dataset Landmark-RxR is proposed to alleviate the weak cross-modal alignment supervision of the coarse-grained data. Besides, inspired by focal loss [23], two kinds of focal-oriented rewards using fine-grained supervision signals are proposed to enhance the local cross-modal alignment by paying more attention to critical points which benefit the navigation process more.

3 Landmark-RxR Dataset

Our Landmark-RxR is built based on the English Guide split of RxR(en-RxR). It contains sub-instruction and sub-trajectory pairs (subpairs) split from instructions in en-RxR. To facilitate the collection of subpairs, we develop a 3D web-based collection tool. It supports the annotator to mark the landmarks in trajectories and split out the corresponding sub-instructions. During the annotation progress, the annotators are allowed to move between the discrete points or change heading and evaluation continuously by mouse click, like what the agent will do in the VLN task. In the following, we will describe the data collection and dataset analysis of Landmark-RxR.

3.1 Data Collection

The annotators are asked to read the complete instruction firstly and then explore the 3D environment following the given trajectory. If the ending points of sub-trajectories, namely landmarks, are found, the annotators need to mark them and split out the corresponding sub-instructions by interacting with the collection tool. As illustrated in Figure 1 (a), the complete instruction and complete trajectory pair (complete pair) is sampled from en-RxR. In Figure 1 (b), three landmarks in the complete trajectory are marked and the corresponding sub-instructions are also split out by annotators. These three sub-instruction and sub-trajectory pairs (subpairs) will be included in our Landmark-RxR dataset. Five principles to ensure the annotation process more standardized are available in the supplemental material.

After an annotator finishes the annotations, another annotator will verify the annotations again and modify the inaccurate part to ensure the annotations satisfy the five principles. For some mistakes in

Figure 2: Agent deviates from the demonstration trajectory after encountering the difficult point. The green points mean the points in the prediction trajectory. The light blue points mean points in the demonstration trajectory. The red triangle represents a difficult point in the trajectory.

RxR, we also correct them in Landmark-RxR. Totally 30 annotators participated in the annotation task, contributing about 2700 hours.

3.2 Dataset Analysis

Table 1: Statistics on R2R, RxR, en-RxR and our Landmark-RxR. edge means the number of edges that each trajectory contains on average. Note that the number of public instructions in RxR test split is 12,469.

dataset	edge	instruction					trajectory
		train	validation seen	validation unseen	test	total	
R2R [4]	5	14,025	1,020	2,349	4,173	21,567	7,189
RxR [15]	8	79,467	8,813	13,625	24,164	126,069	16,522
en-RxR [15]	8	26,464	2,939	4,551	-	33,954	11,321
Landmark-RxR	1.6	133,602	13,591	19,547	-	166,740	46,645

Table 1 gives statistics on R2R, RxR, en-RxR and Landmark-RxR. The total number of sub-instructions from Landmark-RxR is 166,740, which contains 133,602 sub-instructions in train split, 13,591 sub-instructions in validation seen split, and 19,547 sub-instructions in validation unseen split. The average number of edges in a sub-trajectory is 1.6 with 21 words in the corresponding sub-instruction on average. The number of sub-trajectories contained in Landmark-RxR is 46,645. The Landmark-RxR has the largest scale of instructions and trajectories and the minimum granularity among current human-annotated datasets. The fine-grained annotations are essential to cross-modal alignment learning and fine-grained evaluation. Because of the minimum granularity property, the dataset can be easily expanded to a larger scale with different granularities by recombining the subpairs.

4 Re-initialization Mechanism and Evaluation Metrics

4.1 Re-initialization mechanism

An agent needs to navigate following the instructions strictly. So it is essential to evaluate the fidelity between predicted trajectories and the instructions exactly. Current metrics like Coverage weighted by Length Score (CLS) [16] and the normalized Dynamic Time Warping (nDTW) [17] try to obtain the fidelity by computing the similarity between prediction trajectory and the demonstration trajectory. However, the nDTW and CLS scores are sensitive to the positions of difficult points. As illustrated in Figure 2, the agent is firstly initialized at point A and ready to follow the given instruction navigating to point B. Then during navigation, the agent deviates from demonstration trajectory seriously after encountering the difficult point which will result in a low fidelity score. The difficult point means the point where the next action is hard to choose because of the difficult visual scene, *e.g.*, cluttered and similar objects in Figure 2, or ambiguous expressions in an instruction. For the same agent,

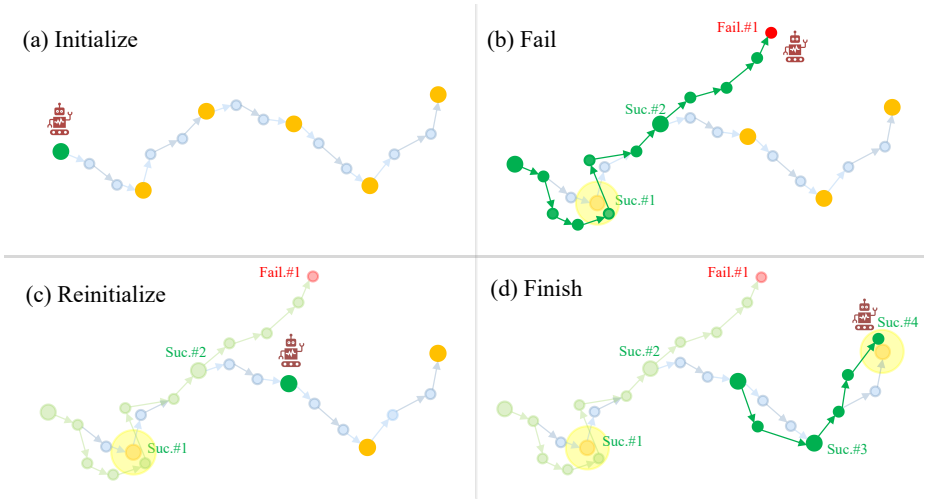

Figure 3: Re-initialization Mechanism. The green points mean the points in the prediction trajectory. The light green points mean prediction trajectory before the last re-initialization and dark green points mean prediction trajectory after the last re-initialization. The light blue points mean points in the demonstration trajectory. The orange points mean the landmarks in the demonstration trajectory. The red point represents the failure point where to re-initialize the agent.

when navigating from point B to point A, because the difficult point is near the end point A, it is easy for the agent to finish the majority of the trajectory and get a higher fidelity score. Therefore, the distributions of difficult points in trajectory have significant impacts on current fidelity metrics, especially for the long trajectories.

Based on the landmark annotations in Landmark-RxR, we introduce the re-initialization mechanism to alleviate the influence of difficult points sensitivity of the fidelity metrics. We define a trajectory as $T = [ST_1, ST_2, ..., ST_{|T|}]$, where $ST_i = [P_{i1}, P_{i2}, ..., L_i]$, and corresponding instruction is defined as $I = [SI_1, SI_2, ..., SI_{|I|}]$. P_{ij} means the jth point in the ith sub-trajectory ST_i. L_i means the ith landmark in the trajectory, in other words, the end point of the ith sub-trajectory and SI_i means the ith sub-instruction.

As illustrated in Figure 3, if the agent succeeds to reach within 3 meters, which is a fixed threshold for success metric defined in [4], of the landmark $L_i(i < |T|)$, or the agent can stop within 3 meters of the landmark $L_{|T|}$ in π (π is set to 10 in this paper) steps, the navigation to landmark L_i is successful. The failure is defined that the agent fails to get within 3 meters of the landmark $L_i(i < |T|)$ in π steps or stops 3 meters away from the landmark L_n. If the agent fails at landmark $L_{|T|}$, we re-initialize the agent at the landmark $L_i(i < |T|)$ as the new start point, and reset the given instruction as $I_{reset} = [SI_{i+1}, ..., SI_{|I|}]$.

Due to the re-initialization mechanism, serious deviation can be corrected in time. We choose fidelity metric nDTW to calculate navigation accuracy of each sub-trajectory. So for a complete trajectory, no matter where the difficult point locates in it, there will be much less influence on averaged navigation accuracy of the sub-trajectories in the complete trajectory.

4.2 Evaluation Metrics

Based on the re-initialization mechanism, Sub-Trajectory Accuracy(SA) and Success weighted by Sub-Trajectory Accuracy(SSA) are defined to measure the sub-trajectory navigation accuracy, and Loss Number(LN) is defined to measure navigation robustness.

Sub-Trajectory Accuracy This metric is defined as nDTW of the predicted sub-trajectory to measure the navigation accuracy of each sub-trajectory.

Success weighted by Sub-Trajectory Accuracy This metric is defined as SDTW of the predicted sub-trajectory.

Loss Number A robust agent should make as few mistakes as possible during navigation. This metric is defined to measure the navigation robustness. Every time the agent fails to navigate to next landmark, the Loss Number adds one. The total numbers of landmarks in validation seen split and validation unseen split of Landmark-RxR are 13,591 and 19,547, respectively.

5 Our Method

We implement our baseline with an encoder-decoder architecture similar to the RCM [6] implemented in [14], where the words in instructions are encoded sequentially but in reverse order originally. Please refer to supplemental material for more details about the baseline.

To enhance the local alignment between instructions and the corresponding critical points, we propose two kinds of focal-oriented rewards benefiting from the fine-grained supervision in Landmark-RxR. The focal-oriented rewards emphasize the critical points in trajectories, which are more helpful for agent to identify navigation directions because of their more detailed descriptions in instructions. Existing rewards are commonly based on either fidelity metric nDTW or success metric SR. To fully demonstrate the key role of critical points in reward shaping, we choose to modify these two metrics to the focal-oriented rewards, namely soft focal-oriented reward and hard focal-oriented reward respectively. Since the landmarks in Landmark-RxR naturally meet the requirements of critical points, we just sample the critical points for each trajectory from its landmark set annotated in our fine-grained Landmark-RxR.

Soft focal-oriented reward In focal-oriented reward, all points in demonstration trajectories are taken into consideration. We base the soft focal-oriented reward on the nDTW metric, which is used to identify the optimal alignment between prediction and demonstration trajectory in time order with minimized cumulative distance. Then we add an importance factor λ to modify the nDTW metric. The importance factor λ gives higher weight to the distances related to the critical points, so it can bias the prediction trajectories in a soft manner, making them close to the demonstration trajectories at positions of critical points especially.

$$R_{focal}^{soft}(R,Q) = exp(-\frac{min_W \Sigma_{(i_k,j_k)\in W} d(r_{i_k}, q_{j_k}) \cdot \lambda_{i_k}}{|R| \cdot d_{th}}) \tag{1}$$

$$\lambda_{i_k} = \begin{cases} 10, & r_{i_k} \in S_R \\ 1, & others \end{cases} \tag{2}$$

where W is a warping of the prediction trajectory and demonstration trajectory defined in [17], d is a distance function mapping pairs of points to a non-negative number, $R = \{r_1, r_2, ..., r_{|R|}\}$ means the demonstration trajectory consisting of point r_i, $Q = \{q_1, q_2, ..., q_{|Q|}\}$ means the predicted trajectory consisting of point q_i and $S_R = \{C_1, C_2, ..., C_{|S_R|}\}$ means the set of sampled critical point C_i in demonstration trajectory. By contrast, the λ is fixed to 1 in fidelity-oriented reward. So in the soft focal-oriented reward, the agent is penalized more once it fails to reach the critical points.

Hard focal-oriented reward In hard focal-oriented reward, only the sampled critical points are considered in the final reward. If the agent succeeds to reach the sampled critical points, the reward is positive to encourage serious decisions during navigation. Otherwise the reward is set as negative value for penalization. The hard focal-oriented reward function is defined as:

$$R_{focal}^{hard} = \Sigma_{C\in S_R} \mathbb{I}(min_{q\in Q}(d(q,C) < d_{th})) \tag{3}$$

$$\mathbb{I}(x) = \begin{cases} 1, & x\ is\ True \\ -1, & x\ is\ False \end{cases} \tag{4}$$

The hard focal-oriented reward encourages the agent to reach critical points during navigation and gives the agent more flexibility in exploring the trajectories between the critical points.

Policy gradient We get loss from the non-differentable reward as Equation 5:

$$L(\theta) = -\Sigma_t logp(a_t|s_t, \theta) \cdot (Re(a_t, s_t) - Re_{avg}) \tag{5}$$

where a_t is the chosen action, s_t is the visual and textual state, θ is the parameters of the network, p is the probability of choosing the action a_t, Re is the reward and Re_{avg} is the averaged reward. Then the gradients to the network parameters are computed as:

$$\nabla_\theta L(\theta) = -\Sigma_t (Re(a_t, s_t) - Re_{avg}) \nabla_\theta logp(a_t|s_t, \theta) \tag{6}$$

6 Experiment

6.1 Setup

Data For experiments on Landmark-RxR, in training phrase, the data contain two parts with different granularities: the sub-instruction and sub-trajectory pairs (subpairs) from Landmark-RxR and the synthesized instruction and synthesized trajectory pairs (synthesized pairs), the augmented data obtained by concatenating several continuous subpairs like [16]. During validation, the agent navigates based on the complete instructions same to en-RxR at the first re-initialization and the instructions synthesized by unfinished sub-instructions after re-initialization. For experiments on en-RxR, the training and validation data contain the complete instruction and complete trajectory pairs (complete pairs) from the en-RxR. For experiments on R2R, the training data contain the subpairs from Landmark-RxR, the complete pairs from en-RxR and the instruction and trajectory pairs from R2R. In validation phrase, due to the lack of high-precision landmark-based data in R2R, the agent is tested on the instructions from R2R validation split without re-initialization mechanism.

Training policy The agent reasons in one stage and end-to-end manner. We warm the agent up by imitation learning with student-forcing. Then we switch to reinforcement learning to learn a more generalizable policy to unseen environment. During reinforcement learning phrase, we adopt the focal-oriented reward function and policy gradient [24] to update parameters in network. Model training consumes about 1,600 minutes in the stage of imitation learning and 3,400 minutes in the stage of reinforcement learning on a single GTX3090 GPU.

Evaluation metrics To fully demonstrate the effectiveness of fine-grained data and the focal-oriented reward function, For R2R and en-RxR datasets, we adopt Success Rate (SR), the percentage of stopping within 3 meters of the final goal, normalized Dynamic Time Warping (nDTW), measuring the fidelity between predicted trajectory and the demonstration trajectory, Success weighted by normalized Dynamic Time Warping (sDTW), measuring the success rate and trajectory fidelity meanwhile. For Landmark-RxR dataset, we adopt Sub-Trajectory Accuracy (SA) and Success weighted by Sub-Trajectory Accuracy (SSA), difficult points insensitive metrics measuring the navigation accuracy, and Loss Number(LN), a difficult points insensitive metric measuring the navigation robustness.

Hyper parameters We use the visual feature from a ResNet-152 [25] trained on ImageNet and adopt panoramic action space like [5]. The word embedding is initialized by GloVe300 [26]. The importance factor λ in soft focal-oriented reward is set to 10. For the focal-oriented reward, we sample same number of critical points in each trajectory to regularize the range of reward value of each episode. The maximum navigation step π allowed for each sub-trajectory is set to 10. R^{soft}_{focal} and R^{hard}_{focal} both sample 2 critical points from landmark set in each trajectory. We set batch size to 100. The total iteration is 100,000 for imitation learning and 20,000 for reinforcement learning.

6.2 Results and Analysis

Fine-grained data contribute to instruction domain generalization Table 2 shows the performance of model#4 is equivalent to model#1 in unseen environment, which indicates the complete pairs in en-RxR have little help for the navigation task in R2R because of the instruction domain gap, which is firstly found in [15]. Once we replace the complete pairs of en-RxR to subpairs of Landmark-RxR, the SR of model#6 improves 7.1% in unseen environment compared to model#1. The better instruction domain generalization indicates that extra cross-modal alignment from the coarse-grained data is weak, but stronger supervision could be obtained from fine-grained data.

Table 2: Multitask [15] results on R2R validation split. *sub* means the subpairs from Landmark-RxR. *co* means the complete pairs from en-RxR. Subpairs from Landmark-RxR help agent improve performance obviously, but the complete pairs from en-RxR have little help.

				Validation Seen				Validation Unseen			
#	R2R	*sub*	*co*	nDTW↑	sDTW↑	SR↑	SPL↑	nDTW↑	sDTW↑	SR↑	SPL↑
1	√			56.7	44.0	53.9	45.1	31.2	19.1	33.0	24.5
2		√		44.6	23.6	31.1	24.8	35.1	15.0	21.6	15.4
3			√	49.7	34.0	43.7	36.1	32.4	15.8	23.6	16.7
4	√		√	65.3	54.5	63.0	56.1	34.3	22.4	33.1	23.0
5	√	√	√	**65.7**	**55.6**	**66.6**	**57.4**	39.5	27.0	38.4	28.1
6	√	√		63.4	53.1	64.1	55.5	**40.3**	**27.5**	**40.1**	**29.3**

Table 3: Results on en-RxR and Landmark-RxR validation seen and validation unseen splits. The two focal-oriented rewards outperform the goal-oriented reward (model#15) and the fidelity-oriented reward (model#16) on almost all metrics.

	IL Train			RL Train				Validation Seen						Validation Unseen					
#	*sub*	*syn*	*co*	R_{focal}^{soft}	R_{focal}^{hard}	R_{nDTW}	R_{SR}	SA↑	SSA↑	LN↓	nDTW↑	sDTW↑	SR↑	SA↑	SSA↑	LN↑	nDTW↑	sDTW↑	SR↑
7	√							56.6	51.7	4337	26.5	12.8	20.4	49.4	44.1	7849	21.8	10.2	18.6
8		√						59.1	56.5	3267	41.7	29.0	42.2	49.7	46.2	7029	24.6	13.8	24.2
9			√					57.6	54.5	3566	44.1	29.9	40.9	48.0	43.8	7452	24.3	12.4	20.7
10	√	√						60.6	58.5	2899	42.1	30.1	43.8	50.4	47.3	6677	24.9	14.9	26.1
11	√		√					60.9	58.6	2899	45.8	34.2	48.2	50.5	47.1	6752	25.1	15.5	26.2
12		√	√					60.6	58.2	3039	47.6	35.3	48.6	49.0	45.3	7302	23.9	13.6	22.8
13	√	√	√					61.8	59.9	2589	46.9	36.0	50.6	51.7	48.9	6256	26.7	17.3	28.4
14	√	√	√			√		64.3	59.8	2764	55.0	32.8	40.7	**58.4**	52.6	5500	**43.2**	22.6	29.3
15	√	√	√				√	61.6	58.8	2661	50.7	36.3	48.2	55.5	51.7	5367	38.2	23.9	32.7
16	√	√	√			√	√	63.0	59.5	2622	54.4	39.3	49.9	57.2	52.5	5309	41.0	25.0	33.6
17	√	√	√	√				**63.8**	**60.7**	2423	**55.1**	**40.7**	**51.4**	57.6	**53.3**	5173	41.0	**25.2**	33.7
18	√	√	√		√			60.7	58.6	**2360**	48.5	37.0	50.7	54.7	51.8	**4951**	36.6	24.2	**34.9**

Fine-grained data contribute to unseen environment generalization As shown in Table 3, trained on synthesized pairs, model#8 outperforms model#7, model#9 and even model#12 in unseen environment on all metrics. It implies that fine-grained data with suitable granularity boost the performance most. Since our Landmark-RxR follows minimum granularity principle during annotating process, our dataset can be easily augmented to various granularities with method like [16] to improve the navigation performance.

When we couple the complete pairs and subpairs as training data (model#11), we see that the LN reduces 700 times and SR improves 5.5% with SSA improving 3.5% meanwhile compared to model#9. Trained on the joint granularity of subpairs, synthetic pairs and complete pairs, model#13 outperforms models#7-12 on all metrics in unseen environment. These indicate cross-modal alignment supervision from coarse-grained data is not enough but the supervision from fine-grained data and coarse-grained data can complement each other to enhance the unseen environment generalization ability.

Focal-oriented rewards facilitate local cross-modal alignment We incorporate different rewards at reinforcement learning phrase, with results recorded in Table 3. Although model#14 which takes nDTW as reward has the highest fidelity scores in unseen environment, it gets the lowest LN and SR scores among all rewards. This indicates that only considering the global alignment between instructions and trajectories make agent just concern the trajectories similarities but not the locations during navigation that instructions really concern.

Compared to model#14, model#17 (soft focal-oriented reward) also bases on nDTW but modifies it with more concern to the critical points. As illustrated in Table 3, the model#17 outperforms the model#14 by a significant margin with SSA increasing from 52.6 to 53.3, LN decreasing from 5500 to 5173 and SR increasing from 29.3% to 33.7%. Besides, model#17 also keeps a slight advantage over model#16 with SSA increasing from 52.5 to 53.3 and LN down from 5309 to 5173, about a 0.7% drop of the total sub-trajectories number 19,547.

We replace the soft focal-oriented reward to the hard formation (model#18). Compared with the model#15 which also using SR[1] as reward signal but only on goal point, SR of model#18 increases from 32.7% to 34.9% and LN of model#18 decreases 416 times, 2.1% of the total number. Meanwhile. Compared with the model#16, the model#18 has 353 decreased in LN during navigation, a 1.8% drop of the total number 19,547, and the SR increasing from 33.6% to 34.9%. As illustrated in Figure

[1] Since trajectories in RxR are not necessary the global shortest to the goal, the reduced distance related part is not included in the goal-oriented reward here.

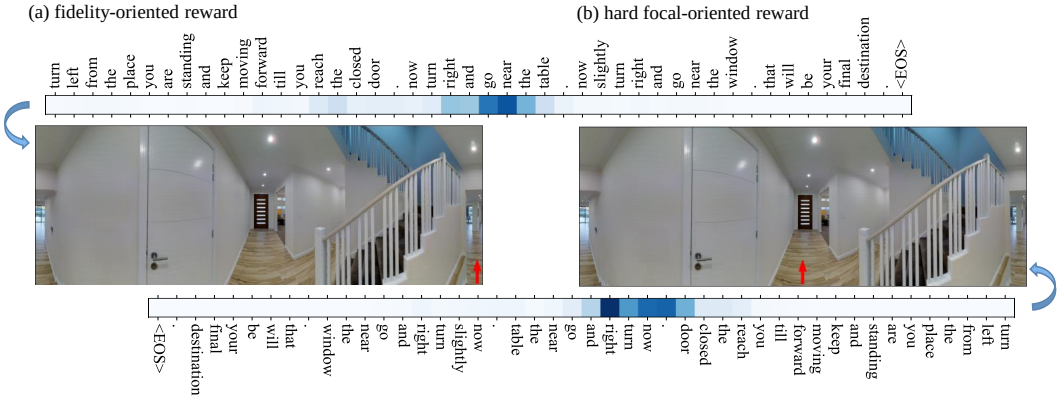

Figure 4: Compare agents trained on fidelity-oriented reward and hard focal-oriented reward. Agent trained on fidelity-oriented reward fails when choosing candidate action to close door, but agent trained on hard focal-oriented reward succeed to align the instruction to the right candidate action.

Table 4: Comparison of performance with the state-of-the-art methods on en-RxR. RCM^{rxr} is the RCM[6] model trained on en-RxR implemented in [15]. Ours^{soft} and Ours^{hard} are the same models to model#17 and model#18, respectively.

		Validation Seen		
#	Agent	nDTW↑	sDTW↑	SR↑
19	RCM^{rxr} [15]	**41.3**	20.3	25.6
21	Ours^{soft}	41.0	**25.2**	33.7
22	Ours^{hard}	36.6	24.2	**34.9**

4, model#18 gives more attention to critical point related words like $close\ door$, and chooses the correct action, which indicate that model#18 performs local cross-modal alignment better.

Above experiments suggest that our focal-oriented rewards which focus on local cross-modal alignment help agent better understand what instructions concern and make fewer mistakes during navigation.

Comparison with SoTA As shown in Table 4, on the en-RxR benchmark, our method with soft focal-oriented reward outperforms RCM^{rxr} obviously on validation unseen split[2], with 4.9 and 8.1% improvement on sDTW and SR metrics respectively. Besides, our method with hard focal-oriented reward achieves the best result on SR metric with 34.9% on validation unseen split, 9.3% absolute improvement compared to RCM^{rxr}.

7 Conclusion

Strong cross-modal alignment ability is the guarantee of successful navigation for a VLN agent. In this paper, we first proposed a human-annotated, fine-grained dataset Landmark-RxR. Our Landmark-RxR provides fine-grained supervision which can complement the weak cross-modal alignment supervision from coarse-grained data. Then benefiting from the fine-grained annotations, two kinds of focal-oriented rewards are proposed focusing on the local cross-modal alignment learning. Besides, the re-initialization mechanism is proposed for difficult points insensitive evaluation on Landmark-RxR. Experiments show that our fine-grained data and focal-oriented rewards help the agent gain superior navigation performance. However, this work is only a preliminary exploration based on our fine-grained dataset Landmark-RxR. We believe that there is much room for further investigation like data augmentation, curriculum reinforcement learning and pretraining. Besides, our fine grain related work like focal-oriented rewards can not only enlightens for VLN task, but also for research on other tasks that can be modeled as Markov decision processes.

[2] Note that monolingual agent is not allowed to test on RxR competition platform. We compare our methods to the state-of-the-art method only on validation unseen split of en-RxR.

References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[2] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *International Conference on Computer Vision*, pages 4534–4542, 2015.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision*, pages 2425–2433, 2015.

[4] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[5] 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. *Advances in Neural Information Processing Systems*, 2018.

[6] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019.

[7] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *Empirical Methods in Natural Language Processing*, 2019.

[8] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *International Conference on Learning Representations*, 2019.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[10] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020.

[11] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, pages 259–274. Springer, 2020.

[12] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[13] Yicong Hong, Cristian Rodriguez-Opazo, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. *Empirical Methods in Natural Language Processing*, 2020.

[14] Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. Babywalk: Going farther in vision-and-language navigation by taking baby steps. *Association for Computational Linguistics*, 2020.

[15] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *Empirical Methods in Natural Language Processing*, 2020.

[16] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. *Association for Computational Linguistics*, 2019.

[17] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *NeurIPS Visually Grounded Interaction and Language Workshop*, 2019.

[18] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *European Conference on Computer Vision*, pages 37–53, 2018.

[19] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[20] Shuhei Kurita and Kyunghyun Cho. Generative language-grounded policy in vision-and-language navigation with bayes' rule. In *International Conference on Learning Representations*, 2021.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[23] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, PP(99):2999–3007, 2017.

[24] Ronald, J., and Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see the data splits in Section 3.2 and other training details in Section 6.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported because it is too computationally expensive.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer the experiment setup part in Section 6.1
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]