A Follower-aware Speaker Model For Vision-and-Language Navigation

Zi-Yi Dou, Nanyun Peng
Department of Computer Science
University of California, Los Angeles
{zdou,violetpeng}@cs.ucla.edu

Abstract

The speaker-follower models have proven to be effective in vision-and-language navigation, where a speaker model is used to synthesize new instructions to augment the training data for a follower navigation model. However, in many of the previous methods, the generated instructions are not directly trained to optimize the performance of the follower. In this paper, we present a follower-aware speaker model that is constantly updated given the follower feedback, so that the generated instructions can be more suitable to the current learning state of the follower. Specifically, we optimize the speaker using a bi-level optimization framework and obtain its training signals by evaluating the follower on labeled data. Experimental results on the Room-to-Room and Room-across-Room datasets demonstrate that our methods can outperform strong baseline models across settings. Analyses also reveal that our generated instructions are of higher quality than the baselines.[1]

1 Introduction

The task of vision-and-language navigation (VLN) requires an agent to navigate in a real-world environment given natural language instructions. In VLN, one of the major challenges is the lack of training data. To alleviate the issue, speaker-follower models (Fried et al., 2018b) have been proposed. Specifically, in the speaker-follower models, an instruction-follower agent is trained to follow a provided natural language instruction to complete a specified goal, and a speaker model learns to model how humans describe routes and synthesize new instructions so as to create more training data for the follower.

While leveraging speaker-augmented data is widely adopted in VLN (Fried et al., 2018b; Wang et al., 2019; Ma et al., 2019; Tan et al., 2019; Zhu et al., 2020a; Hao et al., 2020; Wang et al., 2021; Chen et al., 2021), most of the previous methods focus on improving the follower navigation model. In contrast, how to improve the speaker model to generate data of higher quality is underexplored. In this line of research, Fried et al. (2018a) build a pragmatic speaker that can synthesize instructions based on how the follower may interpret the instructions; Tan et al. (2019) propose to randomly add noise into the environments when generating instructions, so that the noisy environments can mimic unseen environments and the generated instructions can be more diverse; Kurita and Cho (2021) propose a generative approach for VLN where a speaker model is trained and the actions of the follower are selected by maximizing the probability of generating the given instruction.

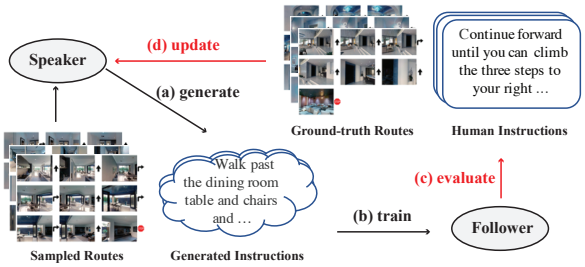

Figure 1: Many of the previous methods use the speaker to generate instructions from sampled routes and train the follower. FOAM (in red) further obtains feedback from the follower on labeled data and updates the speaker accordingly.

In this paper, we propose a follower-aware speaker model (FOAM) that optimizes the generated instructions by directly obtaining feedback from the follower so that the generated instructions can be more suitable for the follower. To this end, we frame the idea as a bi-level optimization problem and obtain the feedback signals to improve the speaker based on the follower performance on labeled data. As illustrated in Figure 1, the follower and speaker are trained in an iterative manner: after updating the follower for one step, it is evaluated

[1]Code is available at https://github.com/PlusLabNLP/follower_aware_speaker.

on a batch of labeled data and the speaker is updated given the performance of the follower. In this way, the speaker is trained to directly optimize the performance of the follower.

Experiments on Room-to-Room (Anderson et al., 2018b) and Room-across-Room (Ku et al., 2020) demonstrate strong performance of FOAM over baselines. Notably, FOAM can achieve comparable performance to a model pre-trained with over millions of text sentences and image-text pairs. Analyses also reveal that our speaker generates instructions of higher qualities than baselines.

2 Methods

We first introduce the background before discussing the details of FOAM.

2.1 Background

Base Settings. VLN requires an agent to follow a given instruction and find a route in a photo-realistic environment (*e.g.* navigate in indoor living spaces). Formally, in an environment $\mathbf{e}$, the follower F parameterized by θ_F learns to model the distribution $P(\mathbf{r}|\mathbf{i};\theta_F)$, where $\mathbf{i}$ and $\mathbf{r}$ denote instruction and route variables, respectively.

The training data $\mathcal{D}$ consists of instruction-route pairs from different environments. Given a batch of instruction-route pairs $(\mathbf{i}_k, \mathbf{r}_k)$ from $\mathcal{D}$, we train the follower F to minimize the cross-entropy loss between its prediction $F(\mathbf{i}_k;\theta_F) = P(\hat{\mathbf{r}}|\mathbf{i}_k;\theta_F)$ and the ground-truth label $\mathbf{r}_k$. Here, we denote this supervised loss as L_l:

$$\min_{\theta_F} L_l(\theta_F) = \mathbb{E}_{(\mathbf{i}_k,\mathbf{r}_k)\sim\mathcal{D}}[\text{CE}(\mathbf{r}_k, F(\mathbf{i}_k;\theta_F)))]. \quad (1)$$

Speaker-Follower Models. Fried et al. (2018b) propose to train a speaker model S parameterized by θ_S that models the distribution of $P(\mathbf{i}|\mathbf{r};\theta_S)$. As in Figure 1, with the speaker, we can perform back translation (Sennrich et al., 2016) on randomly sampled routes $\hat{\mathbf{r}}$ from the training environments $\mathcal{E}$ for data augmentation. Specifically, we first train the speaker S on the same training data as the follower. Then, given a batch of sampled route $\hat{\mathbf{r}}_k \sim \mathcal{E}$, we synthesize their human-like textual instructions $\hat{\mathbf{i}}_k = S(\hat{\mathbf{r}}_k;\theta_S)$. Afterwards, the synthesized training instances $(\hat{\mathbf{i}}_k, \hat{\mathbf{r}}_k)$ are used to update F. Here, we denote this loss as L_u:

$$\begin{aligned}\min_{\theta_F} L_u(\theta_F,\theta_S) &= \mathbb{E}_{(\hat{\mathbf{i}}_k,\hat{\mathbf{r}}_k)\sim\mathcal{E}}[\text{CE}(\hat{\mathbf{r}}_k, F(\hat{\mathbf{i}}_k;\theta_F))] \\ &= \mathbb{E}_{\hat{\mathbf{r}}_k\sim\mathcal{E}}[\text{CE}(\hat{\mathbf{r}}_k, F(S(\hat{\mathbf{r}}_k;\theta_S);\theta_F))].\end{aligned} \quad (2)$$

2.2 Optimizing the Speaker

As we can see from Equation 2, the resulting follower parameters θ_F^* depends on the speaker parameters θ_S, and we can express the dependency as $\theta_F^*(\theta_S)$. However, existing speaker-follower models fail to incorporate θ_S into the optimization process and θ_S is always fixed during training.

Formulation. In this paper, we propose to optimize the parameters of both the follower and speaker during back translation. Specifically, taking inspirations from Pham et al. (2021a,b), we optimize the speaker based on the performance of the follower on the labeled training data, which can be expressed as:

$$\begin{aligned}&\min_{\theta_S} L_l(\theta_F^*(\theta_S)), \\ &\text{where } \theta_F^*(\theta_S) = \operatorname*{argmin}_{\theta_F} L_u(\theta_F,\theta_S).\end{aligned} \quad (3)$$

The motivation of Equation 3 is that while the speaker-augmented data can provide additional supervisions for the follower, the main objective of the speaker is to make the follower better follow human instructions, thus we should focus on minimizing follower's loss on the labeled training data.

Approximation. Following previous work in bi-level optimization (Finn et al., 2017; Liu et al., 2018; Pham et al., 2021a,b), we can approximate argmin with one-step gradient update and alternatively update the parameters θ_F and θ_S.

Specifically, at training step t, we first sample a batch of routes and synthesize their instructions using the speaker S. The generated data is used to update the follower:

$$\theta_F^t = \theta_F^{t-1} - \eta_F \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1}), \quad (4)$$

where η_F is the learning rate.

Then, the speaker is updated to optimize the objective $L_l(\theta_F^t))$ with

$$\theta_S^t = \theta_S^{t-1} - \eta_S \nabla_{\theta_S} L_l(\theta_F^t(\theta_S)). \quad (5)$$

We can approximate the gradient $\nabla_{\theta_S} L_l(\theta_F^t))$ (derivation details in Appendix A) with

$$-[\nabla_{\theta_F} L_l(\theta_F^t)^{\mathrm{T}} \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1})] \nabla_{\theta_S} \log P(\hat{\mathbf{i}}_k|\hat{\mathbf{r}}_k;\theta_S^{t-1}). \quad (6)$$

We can see that this equation resembles REINFORCE (Williams, 1992) in reinforcement learning. Therefore, this algorithm can also be interpreted as treating the similarity in the gradients of the follower model on the labeled data and on the augmented data as rewards, and update the speaker model using reinforcement learning.

Model	Val-Seen			Val-Unseen			Test		
	SR↑	SPL↑	NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑	NE↓
Previous Work									
EnvDrop-ResNet (Tan et al., 2019)	62.1	59	3.99	52.2	48	5.22	51.5	47	-
AuxRN (Zhu et al., 2020a)	70	**67**	3.33	55	50	4.71	55	51	5.15
RelGraph (Hong et al., 2020)	67	65	3.47	57	53	4.73	55	52	4.75
EnvDrop-CLIP-ResNet (Shen et al., 2022)	-	-	-	-	-	-	59.2	53	-
Our Implementations									
Base Follower-CLIP-ViT	60.5	56.6	3.97	54.9	49.3	4.81	-	-	-
EnvDrop-CLIP-ViT	66.1	61.7	3.61	59.2	52.4	4.31	60.0	53.9	4.38
FOAM-CLIP-ViT	**70.8**	**66.6**	**3.25**	**61.6**	**55.1**	**4.18**	**62.2**	**56.2**	**4.09**

Table 1: Results on Room-to-Room. We report success rates (SR), success rates weighted by path length (SPL), navigation error (NE). The best scores are in **bold**. We implement the models based on CLIP-ViT which is stronger than ResNets (row 6 vs. row 1/4). 'Base Follower' is our follower model pre-trained without using the speaker-augmented data. 'EnvDrop' is the best existing speaker-follower baseline.

End-to-End Reconstruction Loss. In this paper, we also propose to add a reconstruction loss for the speaker. Concretely, we compute the gradient of Equation 2 with respect to the speaker parameter θ_S using straight-through estimator, denoted as $\nabla_{\theta_S} L_u(\theta_F, \theta_S)$, and then update the speaker in an end-to-end manner.

To sum up, in FOAM, the final gradient of the speaker is computed based on both the reconstruction loss (Equation 2) and the bi-level optimization loss (Equation 6), and we will perform ablations on the two objectives in the experiment section.

3 Experiments

In this section, we first describe our experimental setup, and then present the results.

3.1 Experimental Setup

Datasets. We evaluate the models on the Room-to-Room (R2R) (Anderson et al., 2018b) and Room-across-Room (RxR) (Ku et al., 2020) datasets. The R2R dataset consists of 7,189 paths, and each path has 3 English instructions with an average length of 29. R2R is split into training, validation, and test sets. The validation set is split into *val-seen*, where paths are sampled from environments seen during training, and *val-unseen*, where paths are sampled from environments that are not seen during training. The paths in the test set are from new environments unseen in the training and validation sets. The RxR dataset follows the same environment division as R2R and there are 16,522 paths in total. The instructions have an average length of 78 and are in three languages, including English, Hindi, and Telugu.

Evaluation Metrics. Our primary metric is success rate (SR), and we also report navigation error (NE), success rate weighted by path length (SPL) on R2R. Following the suggestion in Ku et al. (2020), we also report normalized dynamic time warping (nDTW) and success rate weighted by dynamic time warping (sDTW) (Magalhães et al., 2019) on RxR.

Implementation Details. Following EnvDrop (Tan et al., 2019),we build our speaker and follower based on LSTM (Hochreiter and Schmidhuber, 1997) and environmental dropout is used during back-translation. The follower is pre-trained with imitation and reinforcement learning, and the speaker is pre-trained with maximum likelihood training. Here, we refer to this pre-trained follower as *base follower*. The two models are pre-trained for 80k steps on R2R and 200k steps on RxR, and then trained with our method until the 300k-th iteration. We perform environmental dropout during training as in Tan et al. (2019), and also use their 176,776 paths randomly sampled from seen environments for back translation. Different from Tan et al. (2019), we use CLIP-ViT-224/16 (Radford et al., 2021) to extract vision features as CLIP vision encoders can be beneficial for VLN models (Shen et al., 2022) and we demonstrate that using CLIP vision encoder can obtain better performance than ResNet-based models in the following parts. We compute the cosine similarities between gradients for Equation 6 following Pham et al. (2021b,a) and also perform the same weighting for the reconstruction loss. Each training takes about 3

Model	Val-Unseen-English				Val-Unseen-Hindi				Val-Unseen-Telugu				Test			
	SR↑	SPL↑	sDTW↑	nDTW↑	SR↑	SPL↑	sDTW↑	nDTW↑	SR↑	SPL↑	sDTW↑	nDTW↑	SR↑	SPL↑	sDTW↑	nDTW↑
Base	40.7	36.4	33.5	52.8	**46.8**	41.5	38.5	56.1	42.6	38.3	35.1	54.6	39.1	35.2	32.7	**49.7**
EnvDrop	42.4	38.3	35.5	53.9	46.5	41.5	38.5	56.0	44.4	39.3	36.5	**54.8**	**41.2**	**36.3**	**33.6**	48.8
FOAM	**42.8**	**38.7**	**35.6**	**54.1**	46.7	**41.8**	**38.6**	**56.5**	**45.6**	**39.7**	**37.0**	54.4	**41.2**	36.2	**33.6**	49.3

Table 2: Results on Room-across-Room. We report success rates (SR), success rates weighted by path length (SPL), success rates weighted by dynamic time warping (sDTW), normalized dynamic time warping (nDTW). The best scores are in **bold**.

Follower	Speaker	Pre-exploration			Beam Search		
		Val-Seen	Val-Unseen	Test	Val-Seen	Val-Unseen	Test
EnvDrop	EnvDrop	66.9	64.2	-	74.9	68.4	-
FOAM	EnvDrop	70.2	66.0	-	77.0	70.6	-
FOAM	FOAM	70.6	66.5	68.4	78.1	72.1	72.2

Table 3: Success rates of different configurations of the speaker-follower models in pre-exploration and beam search settings on Room-to-Room. The best configuration is using both *our* follower and *our* speaker models.

days on 1 NVIDIA V100 GPU to finish. We report numbers of a single run for evaluations.

3.2 Main Results

Room-to-Room. We report the main results on R2R in Table 1. We can see that our implementation of the baseline EnvDrop model is better than the previous work because of the stronger vision encoder we use. Based on the strong baseline, our model achieves further improvements on both validation and test sets, outperforming EnvDrop by 2.2% in the success rate on the R2R test dataset, suggesting that our framework is indeed effective.

Room-across-Room. We report the main results on R2R in Table 2. From the table, we can see that the improvements of our framework are not as good on the RxR dataset, possibly because the instructions are much longer and thus it is hard to train a good speaker. Specifically, we find that the baseline speaker can only achieve a BLEU score of 7.4 on the English validation set on RxR (compared with over 30 BLEU scores on R2R as in Appendix B), which leads to noisy augmented data and can impact the performance of speaker-follower models.

3.3 Analysis

In this section, we perform several analyses to gain insights towards our models.

Pre-exploration and Beam Search. We perform experiments in both pre-exploration and beam search settings following previous work (Tan et al., 2019). Because both the speaker and follower are used in the two settings, the evaluation results can reflect the quality of both of the models. As shown in Table 3, we find that the best configuration is using *our* follower and *our* speaker, suggesting that both our follower and speaker are more suitable for VLN than the baselines. Notably, in the beam search setting, our model without pre-training can achieve a success rate of 72.2%, which is comparable to the performance of VLN-BERT (Majumdar et al., 2020) that achieves a success rate of 73% and is pre-trained with over millions of text sentences and image-text pairs.

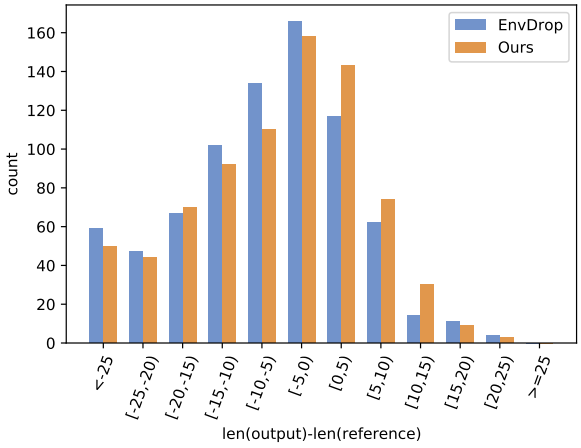

Figure 2: A histogram of the differences in length between the reference and model outputs. Baseline can often generate shorter instructions than the references, but our method can alleviate the issue.

Generated Instructions. The previous pre-exploration and beam search results well indicate

Model	Val-Seen			Val-Unseen		
	SR	SPL	NE	SR	SPL	NE
FOAM	70.8	66.6	3.25	61.6	55.1	4.18
-Recon.	68.9	63.5	3.33	60.2	53.1	4.30
-Bi-level	69.6	65.3	3.33	60.7	54.6	4.27

Table 4: Ablation studies on our proposed objectives. Our reconstruction loss and bi-level optimization loss are complementary to each other and ablating either one of them can lead to degraded performance.

that our generated instructions are more suitable for our follower, suggesting the effectiveness of our framework. In this paragraph, we also compare the generated instructions with the reference instructions. In Figure 2, we plot the histogram of length differences between the reference sentences and the generated instructions using *compare-mt* (Neubig et al., 2019). The figure suggests that the baseline model can often generate shorter instructions than the references, but our method can alleviate this issue, indicating that our methods can indeed improve the speaker quality during training. We also find that our generated instructions are quantitatively and qualitatively better than the baseline using automatic evaluations as in Appendix B.

Ablation Studies. As mentioned in Section 2.2, we perform ablation studies on both of our proposed objectives, namely the bi-level optimization loss (Equation 5) and reconstruction loss. As shown in Table 4, ablating either of the objectives can lead to degraded performance on the R2R validation sets, indicating that both the objectives can improve the model performance and they are complementary to each other.

4 Related Work

We overview two lines of related work:

Vision-and-Language Navigation. Training embodied navigation agents has been an increasingly active research area (Anderson et al., 2018a,b; Chen et al., 2019; Ku et al., 2020; Shridhar et al., 2020; Padmakumar et al., 2022). Fried et al. (2018b) propose to augment the training data with the speaker-follower models, which is improve by Tan et al. (2019) who add noise into the environments so that the speaker can generate more diverse instructions. Zhao et al. (2021) propose methods to measure the quality of the generated instructions and filter noisy samples. Liu et al. (2021) propose to adversarially sample the most difficult paths for the follower and translate these paths into instructions using the speaker for data augmentation. While using the speaker-augmented data has been widely used in VLN, most of the existing work has been focused on improving the follower navigation model (Wang et al., 2018; Li et al., 2019; Zhu et al., 2020b). For example, the self-monitoring agent (Ma et al., 2019) improves cross-modal alignment through a visual-text co-grounding module and a progress monitor; Zhu et al. (2020a) propose to utilize four self-supervised auxiliary tasks that can provide additional training signals for the agent. Most similar to our work, Fried et al. (2018a) build a speaker that reason about how the instructions may be interpreted; Kurita and Cho (2021) propose a generative approach where a speaker model is trained to model the probability of an instructions given actions, and the follower chooses actions that maximize this probability.

Bi-level Optimization. Bi-level optimization algorithms have been widely applied in various fields, such as learning initialization parameters (Finn et al., 2017), neural architecture search (Liu et al., 2018), re-weighting training data (Wang et al., 2020). Our method takes inspirations from (Pham et al., 2021a), which is applied in pseudo labeling and optimizes the teacher parameters given the student feedback. Similar techniques have also been used in machine translation (Pham et al., 2021b), where a meta-validation set is constructed to evaluate the model performance and provide feedback. In this paper, we adopt the techniques in vision-and-language navigation and does not construct a meta-validation set for simplicity.

5 Conclusions

In this paper, we propose the FOAM model where we improve the speaker-follower model in vision-and-language navigation by constantly updating the speaker given the follower feedback during training. We frame the idea as a bi-level optimization problem and obtain the feedback signal based on the performance of the follower on labeled data. Experimental results on Room-to-Room and Room-across-Room datasets demonstrate that our method can outperform strong VLN baselines in different settings. Analyses also suggest that the quality of our speaker model is indeed improved during training. Future directions include testing our method on more datasets and investigating more options on the feedback signals.

Acknowledgement

We would like to thank the anonymous reviewers for valuable suggestions and Te-Lin Wu for helpful discussions. This work is supported by the Machine Common Sense (MCS) program under Cooperative Agreement N66001-19-2-4032 with the US Defense Advanced Research Projects Agency (DARPA).

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. 2018a. On evaluation of embodied navigation agents. *arXiv preprint*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from RGB-D data in indoor environments. In *Proceedings of the International Conference on 3D Vision (3DV)*.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021. History aware multimodal transformer for vision-and-language navigation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018a. Unified pragmatic models for generating and following instructions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. 2020. Language and visual entity relationship graph for agent navigation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shuhei Kurita and Kyunghyun Cho. 2021. Generative language-grounded policy in vision-and-language navigation with bayes' rule. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. 2019. Robust navigation with language pretraining and stochastic sampling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. 2021. Vision-language navigation with random environmental mixup. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Gabriel Magalhães, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. In *Workshop on Visually Grounded Interaction and Language (ViGIL)*.

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) Demonstrations*.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021a. Meta pseudo labels. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hieu Pham, Xinyi Wang, Yiming Yang, and Graham Neubig. 2021b. Meta back-translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. How much can clip benefit vision-and-language tasks? In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. 2021. Structured scene memory for vision-language navigation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. 2020. Optimizing data usage via differentiable rewards. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alexander Ku, Jason Baldridge, and Eugene Ie. 2021. On the evaluation of vision-and-language navigation instructions. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020a. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. 2020b. Babywalk: Going farther in vision-and-language navigation by taking baby steps. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

A Derivation of the Speaker Gradient

As shown in Section 2.2, at training step t, we update the follower according to:

$$\theta_F^t = \theta_F^{t-1} - \eta_F \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1}). \quad (7)$$

Model	Train	Val-Seen	Val-Unseen
BLEU			
EnvDrop	38.16	32.42	31.13
FOAM	39.66	33.11	31.10
BERTScore			
EnvDrop	91.64	91.08	91.04
FOAM	91.79	91.08	91.10

Table 5: Automatic evaluations of the generated instructions. The instructions generated by our model can obtain higher BLEU and BERTScore than the baseline.

We then derive the speaker gradient following previous work (Pham et al., 2021b,a). We define the *expected* parameters of the follower as $\bar{\theta}_F^t$:

$$\bar{\theta}_F^t = \mathbb{E}_{\hat{\mathbf{r}}_k \sim \mathcal{E}, \hat{\mathbf{i}}_k \sim P(\mathbf{i}|\hat{\mathbf{r}}_k; \theta_S^{t-1})}[\theta_F^{t-1} - \eta_F \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1})]. \tag{8}$$

Then, using the chain rule, we can obtain

$$\nabla_{\theta_S} L_l = \frac{\partial L_l}{\partial \bar{\theta}_F^t} \frac{\partial \bar{\theta}_F^t}{\partial \theta_S}, \tag{9}$$

where the first term can be approximated with $\frac{\partial L_l}{\partial \theta_F^t}$. Then, for the second term, we have

$$\frac{\partial \bar{\theta}_F^t}{\partial \theta_S} = \frac{\partial}{\partial \theta_S} \mathbb{E}_{\hat{\mathbf{r}}_k \sim \mathcal{E}, \hat{\mathbf{i}}_k \sim P(\mathbf{i}|\hat{\mathbf{r}}_k)}[\theta_F^{t-1} - \eta_F \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1})]. \tag{10}$$

We can assume that θ_F^{t-1} does not depend on θ_S with Markov assumption (Pham et al., 2021a), and apply the REINFORCE (Williams, 1992) equation on the second term:

$$\begin{aligned} \frac{\partial \bar{\theta}_F^t}{\partial \theta_S} &= \frac{\partial}{\partial \theta_S} \mathbb{E}_{\hat{\mathbf{r}}_k \sim \mathcal{E}, \hat{\mathbf{i}}_k \sim P(\mathbf{i}|\hat{\mathbf{r}}_k)}[-\eta_F \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1})] \\ &= -\eta_F \mathbb{E}_{\hat{\mathbf{r}}_k \sim \mathcal{E}, \hat{\mathbf{i}}_k \sim P(\mathbf{i}|\hat{\mathbf{r}}_k)}[\nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1}) \frac{\partial}{\partial \theta_S} \log P(\hat{\mathbf{i}}_k|\hat{\mathbf{r}}_k; \theta_S^{t-1})], \end{aligned} \tag{11}$$

Using Monte Carlo approximation to approximate terms in Equation 11 using a batch of samples and substituting the result into Equation 9, we can get

$$\nabla_{\theta_S} L_l = -\eta_F [\nabla_{\theta_F} L_l(\theta_F^t)^{\mathrm{T}} \nabla_{\theta_F} L_u(\theta_F^{t-1}, \theta_S^{t-1})] \nabla_{\theta_S} \log P(\hat{\mathbf{i}}_k|\hat{\mathbf{r}}_k; \theta_S^{t-1}), \tag{12}$$

Note that here η_F is a hyper-parameter and can be incorporated into the learning rate of the speaker η_S, thus we remove this term in Section 2.2 and our derivation is complete.

B Evaluations of the Generated Instructions

Automatic Evaluations. As in Table 5, we measure the quality of the generated instructions in BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020). We find that our speaker can generate instructions of higher qualities according to the two metrics.

Method	Instruction
Reference	walk downstairs and outside . stop in the outhouse through the door on the right .
EnvDrop	go down the stairs and turn right . go down the hallway and stop in front of the door .
FOAM	go down the stairs and turn right . go down the hallway and go through the door on the right .
Reference	turn left and take a right at the table . take a left at the painting and then take your first right . wait next to the exercise equipment .
EnvDrop	walk past the dining room table and chairs and turn left . walk past the table and chairs and turn right . walk into the room and stop .
FOAM	walk past the dining room table and chairs and turn left . walk past the table and chairs and turn right . walk into the room and turn right . stop in front of the exercise bike .

Table 6: Examples of the generated instructions. Our generated instructions are generally longer and more accurate compared with the baseline.

Qualitative Examples. As in Table 6, we also find that after training the speaker using our method, the generated instructions are generally longer than the baseline and are more accurate compared with the references.

C License

We evaluate our models on the Room-to-Room (R2R) (Anderson et al., 2018b) and Room-across-Room (RxR) (Ku et al., 2020) datasets based on Matterport3D (Chang et al., 2017). The datasets are released under the Matterport3D Terms of Use.[2] The datasets do not contain any information that names or uniquely identifies individual people or offensive content. Our code is based on EnvDrop that is released under the MIT license.[3] We use the datasets and code for research purposes, which is consistent with their intended use.

D Limitations and Potential Risks

As in the experiments, our models may not work well when the instructions are long and it is hard to train a reasonable speaker model. Also, our model requires fine-tuning the speaker during training based on the feedback of the follower, which

[2] http://dovahkiin.stanford.edu/matterport/public/MP_TOS.pdf

[3] https://github.com/airsplay/R2R-EnvDrop/blob/master/LICENSE

introduces additional training costs to the model. In addition, the datasets we use in the paper may make our model biased towards environments of American buildings.