Vision Language Navigation with Knowledge-driven Environmental Dreamer

Fengda Zhu[1] , **Vincent CS Lee**[1] , **Xiaojun Chang**[2] and **Xiaodan Liang**[3,4]
[1]Monash University
[2]University of Technology Sydney
[3]Sun Yat-sen University
[4]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
{fengda.zhu, vincent.cs.lee}@monash.edu, cxj273@gmail.com, xdliang328@gmail.com

Abstract

Vision-language navigation (VLN) requires an agent to perceive visual observation in a house scene and navigate step-by-step following natural language instruction. Due to the high cost of data annotation and data collection, current VLN datasets provide limited instruction-trajectory data samples. Learning vision-language alignment for VLN from limited data is challenging since visual observation and language instruction are both complex and diverse. Previous works only generate augmented data based on original scenes while failing to generate data samples from unseen scenes, which limits the generalization ability of the navigation agent. In this paper, we introduce the Knowledge-driven Environmental Dreamer (KED), a method that leverages the knowledge of the embodied environment and generates unseen scenes for a navigation agent to learn. Generating an unseen environment with texture consistency and structure consistency is challenging. To address this problem, we incorporate three knowledge-driven regularization objectives into the KED and adopt a reweighting mechanism for self-adaptive optimization. Our KED method is able to generate unseen embodied environments without extra annotations. We use KED to successfully generate 270 houses and 500K instruction-trajectory pairs. The navigation agent with the KED method outperforms the state-of-the-art methods on various VLN benchmarks, such as R2R, R4R, and RxR. Both qualitative and quantitative experiments prove that our proposed KED method is able to high-quality augmentation data with texture consistency and structure consistency.

1 Introduction

Vision-Language Navigation (VLN) task [Anderson *et al.*, 2018a] requires an agent to navigate in an embodied environment following a natural language instruction. This task is closely connected to many real-world applications, such as household robots and rescue robots [Zhu *et al.*, 2021]. The VLN task is challenging since it requires an agent to master diverse skills, such as vision-language alignment, sequential vision perception, and long-term decision-making. The key to the VLN task is to perceive the panoramic visual scene, comprehend natural language instructions sequentially, and make actions step-by-step.

Learning a robust navigation policy is a long-study problem in the Artificial Intelligence community. Previous methods [Fried *et al.*, 2018; Wang *et al.*, 2019; Chen *et al.*, 2021; Qiao *et al.*, 2022] attempt to have made great progress in improving the ability to perceive the vision and language inputs and learn a robust navigation policy [Liu *et al.*, 2021; Hao *et al.*, 2020; Hong *et al.*, 2021]. However, the limited scale of training data introduces a large bias between training environments and testing environments, which severely impacts navigation performance. The most widely used VLN dataset, Room-to-room dataset [Anderson *et al.*, 2018b], contains only 22K instruction-path pairs from 90 house scenes. However, the space of possible navigation paths and visual observations increases exponentially along with the path length. Therefore, the learned navigation policy can easily overfit the seen scenes and is hard to generalize to the unseen scenes. For example, if an agent learns navigation in a scene with red chairs and a blue sofa, it may be confused when it is tested in an unseen scene with blue chairs and a red sofa.

In this paper, we propose a model named Knowledge-driven Environmental Dreamer (KED), which is able to generate unseen environmental scenes without extra data annotations. The environmental dreamer is built based on an auto-encoder model, which consists of an encoder and a decoder. The encoder learns to disentangle two latent encodings from an image view: a texture encoding and a structure encoding. The decoder receives these two latent encodings and generates a synthetic image to represent a panoramic view of an agent in an unseen environment. The environmental dreamer generates an unseen environment by three steps: 1) we divide house scenes into rooms; 2) we extract texture encodings and structure encodings of each room; 3) we randomly match the texture encodings and the structure encodings from different house scenes and decode them to generate unseen house scenes. An overview of our Knowledge-driven Environmental Dreamer method is shown in Figure 1.

Different from previous image generation works [Park *et al.*, 2020; Zhu *et al.*, 2020b; Karras *et al.*, 2021] that synthesize unseen images, our work is required to generate unseen

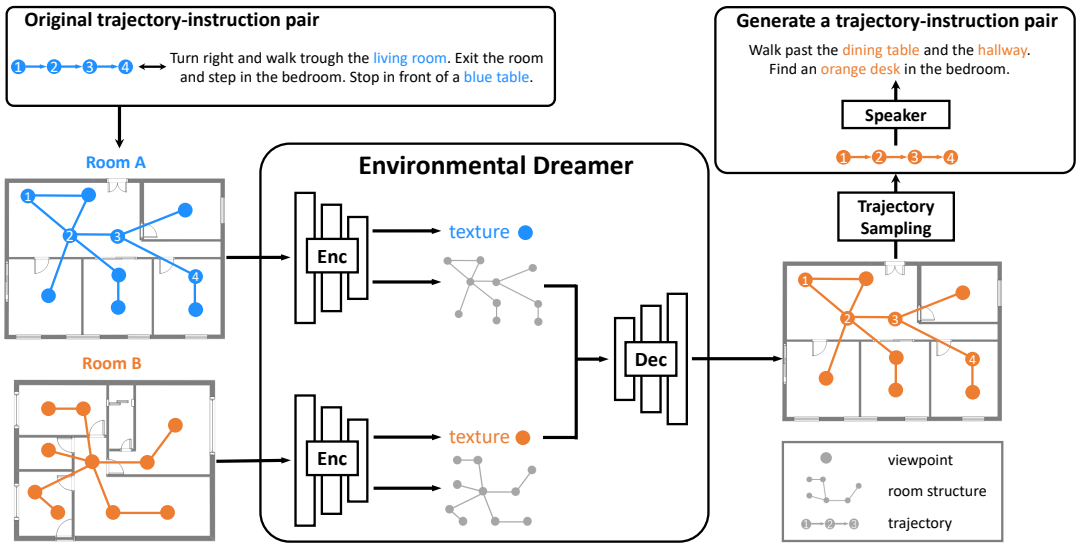

Figure 1: An overview of our proposed knowledge-driven Environmental Dreamer (KED). The environmental dreamer takes two house scenes as input, and uses an encoder to disentangle the texture encoding and semantic encoding of the two scenes. Then it decodes a combination of a texture encoding from one room and a semantic encoding from the other room to generate a new house scene. We introduce a "speaker" model to describe the sampled trajectory in the generated seen as an instruction with new entities to augment our training data.

environments. Generating unseen environments is challenging because we need to ensure texture consistency and structure consistency of the generated environments. The texture consistency indicates that the nearby views should have similar textures. If a sofa could be observed from several positions, the sofa should have the same color in the views of these positions. The structure consistency means the house structure shown in nearby views should be consistent. If a pillar could be observed from several positions, the pillar should have the same shape in the views of these positions. Therefore, we introduce semantic regularization objectives in order to ensure texture consistency and structure consistency. First of all, we suggest that all panoramic views in the same room, which are represented as all nodes in a connected block, should be transferred by the same texture. Second, we propose that the semantic information of an original panoramic view and the generated view should be consistent. Third, we suggest that a trajectory sampled in a generated house scene should contain the same semantic information as the original trajectory.

Our experiments show that the navigation agent with KED outperforms the previous state-of-the-art method on various VLN benchmarks, such as R2R, R4R, and RxR. The results also demonstrate that using the augmentation data generated by KED is able to significantly reduce the performance gap between seen and unseen environments, which dramatically improves the overall navigation performance. Our ablation study shows that the proposed augmentation method outperforms other augmentation methods at the same augmentation data scales. The visualization results show that our proposed regularization objectives are able to ensure texture consistency and structure consistency and generate environments with higher-quality images.

2 Related Work

Embodied Navigation Environments. have been proposed for navigation learning. House3D [Wu *et al.*, 2018] is the first indoor environment for navigation. AI2-THOR [Kolve *et al.*, 2017] is an interactable indoor environment. The Active Vision dataset [Ammirato *et al.*, 2017] consists of dense scans of 16 different houses. Matterport3D [Anderson *et al.*, 2018b], Gibson [Xia *et al.*, 2018; Xia *et al.*, 2020] and Habitat [Savva *et al.*, 2019] propose high-resolution photo-realistic panoramic view. However, the scales of these environments are small since collecting data for rendering house scenes is expensive.

Vision-language Navigation. Anderson *et al.* [Anderson *et al.*, 2018b] propose the Room-to-Room (R2R) dataset, which is the first Vision-Language Navigation (VLN) benchmark [Chang *et al.*, 2017]. To address the VLN task, Fried *et al.* propose a speaker-follower framework [Fried *et al.*, 2018] for data augmentation and reasoning in a supervised learning context, along with a concept named "panoramic action space" that is proposed to facilitate optimization. Wang *et al.* [Wang *et al.*, 2019] demonstrate the benefit of combining imitation learning [Bojarski *et al.*, 2016; Ho and Ermon, 2016] and reinforcement learning [Mnih *et al.*, 2016; Schulman *et al.*, 2017]. Due to the success of BERT [Devlin *et al.*, 2018], researchers have extended it to learn vision-language representations in VLN. PRESS [Li *et al.*, 2019] applies the pre-trained BERT to process instructions. PREVALENT [Hao *et al.*, 2020] pre-trains an encoder with image-text-action triplets to align the language and visual states, while VLN-BERT [Majumdar *et al.*, 2020] fine-tunes ViLBERT [Lu *et al.*, 2019] with instruction-trajectory pairs. Hong *et al.* [Hong *et al.*, 2021] implement a recurrent function to leverage the history-dependent state representations

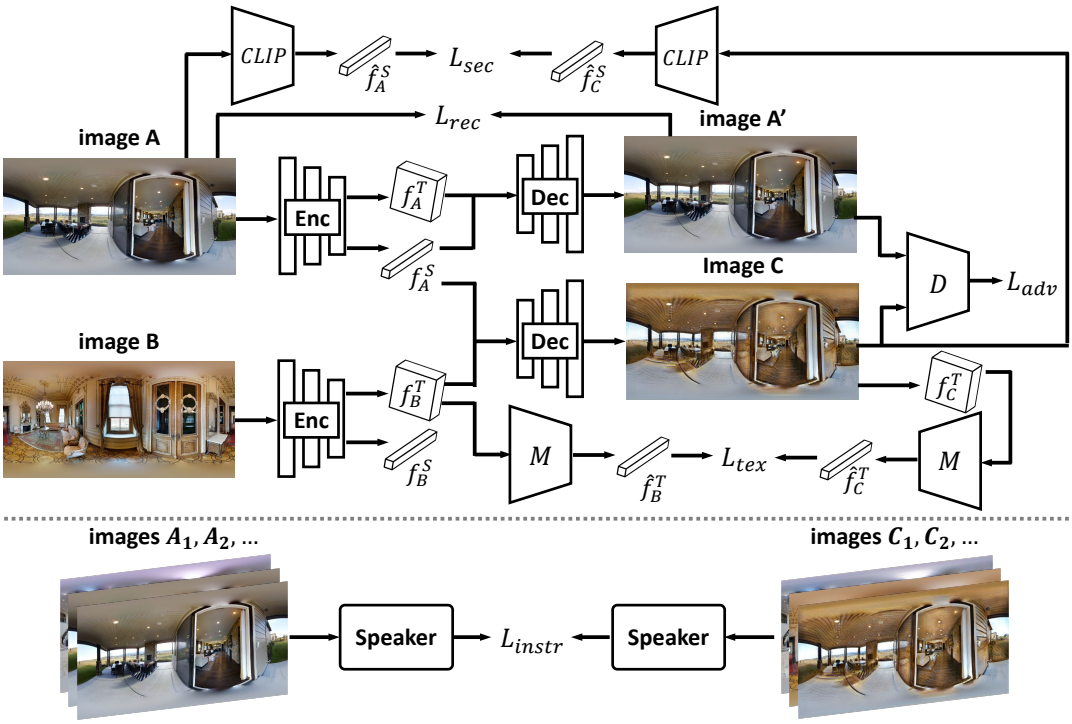

Figure 2: The detailed framework of our Knowledge-driven Environmental Dreamer (KED).

based on previous models. However, the agent learned with limited training scenes can overfit seen textures, and hard to generalize to unseen scenes where objects look different.

Style transfer. requires blending two objects, one containing content and the other containing style, to integrate into a new object. Many classic works of style transfer [Hertzmann *et al.*, 2001; Johnson *et al.*, 2016; Tenenbaum and Freeman, 2000] focus on obtaining human faces of different styles [Hertzmann *et al.*, 2001; Tenenbaum and Freeman, 2000] as well as art works [Chen *et al.*, 2017; Gatys *et al.*, 2016]. Later, with the potentiality of style transfer being revealed, it has been applied to some challenging tasks like navigation [Zhu *et al.*, 2020c] and super-resolution [Johnson *et al.*, 2016]. However, the application of this technique to vision language navigation is not trivial since it is difficult to generate unseen scenes which vary in size, shape, and texture. Compared with previous works that generate a synthetic scene [Zhao *et al.*, 2021] or introduce segmentation data [Koh *et al.*, 2021], our work makes the first attempt at generating photo-realistic scenes without additional prior.

3 Knowledge-driven Environmental Dreamer

3.1 Problem Setup

In the Vision-and-Language Navigation (VLN) task, an agent follows a natural language sentence and predicts actions $A = \{a_1, ..., a_n\}$ step-by-step to reach the target. A navigation environment is formulated as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consisting of nodes and edges. At each node, a navigation agent is able to observe a panoramic image view I. The agent takes an action a to navigate from one node to another via an edge.

3.2 Learning to Dream

Here we introduce the framework and the objective functions for learning the environmental dreamer. The detailed framework is shown in Figure 2. Our environmental dreamer consists of an encoder E and a decoder G. The encoder takes a panoramic view as input and outputs a texture encoding f^T and a semantic encoding f^S. The decode is used to decode a concatenated feature with a texture encoding and a semantic encoding as a panoramic image view.

Reconstruction and Adversarial Objectives. For each image I in a minibatch, we calculate a reconstruction loss as:

$$L_{rec}(E, G) = \|I - G(E(I))\| . \tag{1}$$

In addition, we introduce an adversarial objective with a discriminator D to improve the quality of the generated image:

$$\begin{aligned} L_{adv}(D) &= -\log(D(G(E(I)))) \\ L_{adv}(E, G) &= -\log(1 - D(G(E(I)))) \end{aligned} \tag{2}$$

This objective encourages the discriminator D to distinguish fake image and encourage the generator G to generate high quality image to deceive D.

Decomposition Objective. is adopted to help our model to learn feature decomposition. Different from classical auto-encoder framework, we divide the latent feature encoding f into two parts: a texture encoding f^T and a semantic encoding f^S. The texture encoding is a shallow but wide feature

map and the semantic encoding f^S the deep but narrow feature map. We sample two images I_A and I_B from different house scenes and optimize the decomposition objective:

$$\begin{aligned} &f_A^T, f_A^S = E(I_A), \qquad f_B^T, f_B^S = E(I_B), \\ &I_C = G([f_A^T, f_B^S]), \qquad I_D = G([f_B^T, f_A^S]), \\ &L_{dec}(D) = -\log(D(I_C)) - \log(D(I_D)), \\ &L_{dec}(E, G) = -\log(1 - D(I_C)) - \log(1 - D(I_D)), \end{aligned} \tag{3}$$

where $[\cdot]$ represents concatenation. I_C and I_D are unseen synthetic images generated by G. The decomposition objective enables the encoder to learn to represent texture encoding and semantic encoding respectively. The encoder and decoder are jointly optimized by these objectives.

3.3 Regularizing Environmental Consistency

Texture Consistency Objective. We propose the texture consistency objective to ensure that all panoramic images in the same room are consistent. Firstly, we adopt a neural mapping network M to map the texture encodings of the original image I_B and the generated image I_C as feature vectors that represent the global texture information. We use a cosine similarity loss to maximize the similarity of the global texture vectors:

$$\begin{aligned} &\hat{f}_B^T = M(f_B^T), \qquad \hat{f}_C^T = M(E(I_C)) \\ &L_{tex}(E, G, M) = 1 - \frac{\hat{f}_B^T \cdot \hat{f}_C^T}{\| \hat{f}_B^T \| \| \hat{f}_C^T \|}. \end{aligned} \tag{4}$$

Semantic Consistency Objective. One difficulty in our environmental generation task is to ensure that all objects should be easily distinguished without distortion, especially small objects at a distance. To address this problem, we propose to leverage the knowledge from the visual encoder of the CLIP [Radford *et al.*, 2021] model. We use the CLIP model to encode the original image I_A and the generated image I_C as features, and use Mean Absolute Error (MAE) to penalize the error between the features:

$$\begin{aligned} &\hat{f}_A^S = \mathrm{CLIP}(I_A), \qquad \hat{f}_C^S = \mathrm{CLIP}(I_C) \\ &L_{sec}(E, G) = |\hat{f}_A^S - \hat{f}_C^S|. \end{aligned} \tag{5}$$

The semantic consistency objective encourages the environmental dreamer to generate panoramic views that contain diverse objects with new textures. It solves the problem that the model focuses on style transfer while ignoring the generation of detail objects.

Instruction Consistency Objective. In addition to visual objectives, we suggest that the semantic information contained in the trajectory of a generated scene should be consistent with which of the original scene. Firstly, we train a "speaker" model P using the instruction-trajectory pairs as in [Fried *et al.*, 2018]. The "speaker" model encodes a sequence of image observations following a trajectory and decodes a natural language sentence to describes the image sequence. Secondly, we sample a trajectory from an original room A and a trajectory from a generated room C, and use the "speaker" model to describe the trajectories by natural language sentences and penalize the error by a cross-entropy loss:

$$L_{ins} = -\sum_i w_{A,i} \log(P(w_{C,i} | w_{A_0}, ..., w_{A_{i-1}}, I_{C_i})). \tag{6}$$

The $w_{A,i}$ denotes the ground truth word at position i in room A, and I_{C_i} denotes the view of position i in room C. We regularize the instruction consistency objective in the "teacher-forcing" paradigm. This training paradigm gives the previous ground truth words to the "speaker" model to reduce the accumulating error. The weights of the "speaker" model are fixed in optimizing the instruction consistency objective.

Algorithm 1 Selecting key vertexes

Input: Two environments $\mathcal{G}_A = \{\mathcal{V}_A, \mathcal{E}_A\}, \mathcal{G}_B = \{\mathcal{V}_B, \mathcal{E}_B\}$, encoder E, decoder D

$\tilde{\mathcal{V}}_A = \{v \mid \text{top } 10\% \ v \text{ in } V_A \text{ ordered by BC}(v)\}$;
$\tilde{\mathcal{V}}_B = \{v \mid \text{top } 10\% \ v \text{ in } V_B \text{ ordered by BC}(v)\}$
$R_A = \mathrm{ConnectedBlock}(\mathcal{V}_\mathrm{A} - \tilde{\mathcal{V}}_\mathrm{A})$;
$R_B = \mathrm{ConnectedBlock}(\mathcal{V}_\mathrm{B} - \tilde{\mathcal{V}}_\mathrm{B})$
for r_A in R_A **do**
  $r_B = \mathrm{random}(R_B)$
  $v_B = \mathrm{random}(r_B)$
  $f_B^T, f_B^S = E(v_B)$
  for v_A in r_A **do**
    $f_A^T, f_A^S = E(v_A)$
    $v_C = D([f_B^T, f_A^S])$;
    $\mathcal{V}_C = \mathcal{V}_C \cup \{v_C\}$;
  end for
end for
for v_A in $\tilde{\mathcal{V}}_A$ **do**
  $v_B = \mathrm{random}(\tilde{\mathcal{V}}_A)$
  $f_A^T, f_A^S = E(v_A)$
  $f_B^T, f_B^S = E(v_B)$
  $v_C = D([f_B^T, f_A^S])$
  $\mathcal{V}_C = \mathcal{V}_C \cup \{v_C\}$
end for
return $\mathcal{G}_C = \{\mathcal{V}_C, \mathcal{E}_C\}$

Loss Reweighting for Adaptive Optimization. If the model focuses on regularization too early in the training process, the performance of image translation can be reduced. Therefore, we design a loss reweighing mechanism to encourage the model to focus on the image translation task at the beginning and pay more attention to regularization after it performs the image translation task well:

$$\begin{aligned} &\alpha = 1 - \frac{1}{2}(L_{adv} + L_{dec}), \\ &L_{reg} = \alpha(L_{tex} + L_{sec} + L_{ins}). \end{aligned} \tag{7}$$

Above all, we sum up all objectives to jointly train the environmental dreamer:

$$L_{total} = \lambda_1 L_{rec} + \lambda_2 L_{adv} + \lambda_3 L_{dec} + \lambda_4 L_{reg}. \tag{8}$$

3.4 Generating Training Data for Navigation

To describe our method for generating augmented training data for the vision-language navigation task, we first generate views of house scenes to build environments. Then we

Method	R2R Validation Seen				R2R Validation Unseen				R2R Test Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
Random	9.58	9.45	16	-	9.77	9.23	16	-	9.89	9.79	13	12
Human	-	-	-	-	-	-	-	-	11.85	1.61	86	76
Speaker-Follower [Fried *et al.*, 2018]	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
RCM+SIL [Wang *et al.*, 2019]	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
PRESS [Li *et al.*, 2019]	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
EnvDrop [Tan *et al.*, 2019]	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
AuxRN [Zhu *et al.*, 2020a]	-	3.33	70	67	-	5.28	55	50	-	5.15	55	51
PREVALENT [Hao *et al.*, 2020]	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
RelGraph [Hong *et al.*, 2020]	10.13	3.47	67	65	9.99	4.73	57	53	10.29	4.75	55	52
VLN⟳Bert [Hong *et al.*, 2021]	11.13	2.90	72	68	12.01	3.93	63	**57**	12.35	4.09	63	57
HOP [Qiao *et al.*, 2022]	11.26	**2.46**	**76**	**70**	12.27	**3.79**	**64**	**57**	12.68	**3.87**	**64**	**58**
RCM* [Tan *et al.*, 2019]	10.25	4.91	53.8	50.7	9.38	5.89	46.2	42.5	9.58	5.88	46.4	43.3
RCM+KED	10.20	3.79	64.2	61.4	10.78	5.39	48.6	44.5	9.81	5.67	48.7	45.1
VLN⟳Bert* [Hong *et al.*, 2021]	12.09	2.99	70.7	65.9	12.58	4.02	61.4	55.6	11.68	4.35	61.4	56.7
VLN⟳Bert+KED	11.27	2.58	75.6	**70.9**	12.01	**3.66**	**64.9**	**58.1**	12.80	3.88	**64.3**	**59.4**

Table 1: Comparison of agent performance on **R2R** in single-run setting. * reproduced results in our environment.

Method	R4R Validation Seen						R4R Validation Unseen					
	NE↓	SR↑	SPL↑	CLS↑	nDTW↑	SDTW↑	NE↓	SR↑	SPL↑	CLS↑	nDTW↑	SDTW↑
Speaker-Follower	4.31	59.0	50.8	50.8	49.9	35.5	6.37	43.5	33.6	40.5	38.8	23.3
Envdrop	4.82	55.8	48.6	50.3	49.0	34.6	6.42	43.1	34.1	40.9	39.0	**23.4**
KED (ours)	**4.36**	**63.7**	**52.5**	**51.7**	**51.1**	**36.9**	**6.26**	**44.0**	**34.5**	**41.5**	**39.3**	**23.4**

Table 2: Comparison of agent performance on **R4R** in single-run setting.

sample instruction-trajectory data from the generated house scenes.

As described in the Problem Setup section, we model a navigation environment as a graph. We pick out the key nodes according to the betweenness centrality (BC). A node is considered a critical node if it has a betweenness centrality score in the top 10%. The connected sub-graphs divided by key nodes are regarded as rooms.

We generate a new house scene by generating new views room-by-room in three steps: 1) we randomly sample a scene **A** to provide semantic information and a scene **B** to provide texture information; 2) for each room in scene **A**, we encode a random view from rooms in scene **B** as f_B^T and use this texture encoding to decode all semantic features of that room to generate a new room; 3) for each key node in scene **A**, we use the texture encoding of a random key node in scene **B** to generate a new view for that key node. A detailed demonstration of this process is shown in Algorithm 1. The reason we only use the texture encoding of scene **B** instead of all the scenes in the dataset is that it prevents the textures from changing too much in the new scene. Ideally, our method is able to generate more than 8K new scenes. Due to the limit of computation resources, we generate 270 scenes in practice, 3 times more than the original scenes provided by the R2R dataset.

Next, we randomly sample trajectories in the generated unseen scenes and use the "speaker" model to describe these trajectories to generate instruction-trajectory pairs. At last, we generated 500K instruction-trajectory pairs, 25 times the size of the original data, which greatly enriches our training data.

3.5 Learning to Navigate

We adopt a joint optimization policy that uses imitation learning and reinforcement learning.

Imitation Learning. forces the agent to mimic the behavior of its teacher. Our agent learns from the teacher action a_t^* for each step:

$$L_{IL} = \sum_t -a_t^* \log(p_t), \tag{9}$$

where a_t^* is a one-hot vector indicating the ground truth.

Reinforcement Learning. We implement the A2C algorithm [Mnih *et al.*, 2016] to maximize the total reward of navigation, whose loss function is formulated as:

$$L_{RL} = -\sum_t a_t \log(p_t) A_t, \tag{10}$$

where A_t is a scalar representing the advantage defined in [Mnih *et al.*, 2016].

4 Experiment

Dataset and Environments. We evaluate our navigation agent on three VLN benchmarks: Room-to-Room (R2R), Room-for-Room (R4R), and Room-Across-Room (RxR). The original training set provided by R2R consists of 61 environments and 14,025 instructions. We augment the training dataset by 500K instruction-trajectory pairs from 270 generated unseen scenes.

Method	RxR Validation Seen						RxR Validation Unseen					
	NE↓	SR↑	SPL↑	CLS↑	nDTW↑	SDTW↑	NE↓	SR↑	SPL↑	CLS↑	nDTW↑	SDTW↑
Speaker-Follower	12.11	20.7	18.9	46.2	37.2	17.1	11.34	21.1	19.0	46.4	38.0	17.6
Envdrop	11.66	22.4	20.8	47.2	39.2	18.7	11.12	21.1	19.4	46.8	38.9	17.8
KED (ours)	**11.43**	**23.0**	**22.2**	**48.2**	**40.8**	**19.3**	**10.97**	**21.7**	**19.6**	**47.5**	**39.7**	**18.1**

Table 3: Comparison of agent performance on **RxR** in single-run setting.

Method	R2R Validation Seen				R2R Validation Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
① VLN⟳Bert	12.09	2.99	70.7	65.9	12.58	4.02	61.4	55.6
② VLN⟳Bert+KED w/o L_{reg}	11.34	2.68	73.9	69.3	11.34	4.00	63.5	55.8
③ VLN⟳Bert+KED with L_{reg}	11.02	3.21	68.3	63.9	11.23	3.95	62.0	56.7
④ VLN⟳Bert+KED (270 scenes)	11.68	**2.71**	**73.7**	**68.6**	12.46	**3.92**	**63.1**	**56.9**

Table 4: Ablation study on **R2R** in validation seen and unseen splits.

Evaluation Metrics. We use a large number of metrics for evaluation, such as Trajectory Length (TL), Navigation Error (NE), Success Rate (SR), and Success rate weighted by Path Length (SPL). In the evaluation of R4R and RxR, we compare our method with previous state-of-the-art methods on Coverage weighted by LS (CLS), Normalized Dynamic Time Warping (nDTW), Success weighted by normalized Dynamic Time Warping (sDTW).

Results on VLN Standard Benchmark. In this section, we compare our method with several other representative methods on the standard R2R benchmark. We apply our KED method on two backbone models: a seq-to-seq model with cross-modal attention [Wang *et al.*, 2019], and a recurrent VLN-BERT model [Hong *et al.*, 2021]. Tab. 1 shows that both backbone models trained with our augmentation data outperform the models without augmentation data. Our method improves the SPL performance of the RCM model by 10.7% on validation seen, 2.0% on validation unseen, and 1.8% on test split. Our method improves the SPL performance of the recurrent VLN-BERT model by 5.0% on validation seen, 2.5% on validation unseen, and 2.7% on test split. The KED based on the VLN-BERT model outperforms the previous state-of-the-art model [Qiao *et al.*, 2022] by 0.9% on validation seen, 1.1% on validation unseen, and 1.4% on test split. Above all, the agent built on the recurrent VLN-BERT model and learned with our augmentation data outperforms all previous methods and achieves the state-of-the-art method on the standard R2R benchmark.

Results on R4R and RxR. Here we investigate if our method works on other navigation benchmarks, such as R4R and RxR. The result are shown in Table 2 and Table 3. We compare KED with other augmentation methods: 1) pairwise data augmentation (Speaker-follower); 2) feature-wise data augmentation (Envdrop); 3) and scene-wise data augmentation (KED). Our KED method significantly outperforms the other two augmentation methods on all metrics in the R4R dataset and RxR dataset, which validate the effectiveness of our method.

Ablation of Knowledge-driven Dreamer (KED). In this section, we ablate different versions of our model and compare how much performance improvement the navigation agent is able to achieve by each part. Results are shown in Table 4. By comparing ① and ②, we find that the training data generated by the baseline model is able to improve the navigation performance. We discover that ③ significantly outperforms ②, which infers that the regularization objective largely improves the quality of training data. ③ is trained with 90 scenes. The performance of this model improves if it is trained with more data like ④, which reveals that our data is beneficial and large-scale scene-wise augmentation data can reduce the training-testing domain gap.

Ablation of Speaker. Here, we ablate the impact of the speaker for our data augmentation method based on the RCM backbone, as shown in Table 5. The KED method without a speaker is able to improve the navigation performance on both validation sets. More performance improvement can be get by applying the speaker, which indicates that the speaker is useful in our data augmentation method.

Quality of Generated Scenes. In Figure 3, we compare the generated panoramic views between the model with regularization objectives and the baseline model without regularization objectives. It turns out that our model with regularization objectives have several advantages: 1) they have fewer fragments and "dirty" regions, such as ①, ④, ⑤; 2) the objects are clearer and easier to distinguish, such as ②, ⑥, ⑧; 3) the shadows and light effects are more realistic. For example, the baseline model produces shades of orange in ⑦ when there is no orange object or light source in the view. We conclude that is because the baseline model mistakenly transfer the texture information to the generated image. 4) the generated views contain more detailed objects, especially the small objects in the distance. In ③, ⑥ and ⑧, the objects generated by the baseline model cannot be seen clearly while our model is able to generate detailed objects. 5) The objects in the generated view look more realistic. In ⑦, the clock is transferred to another texture but the numbers are still clear. In contrast, the

Method	R2R Validation Seen				R2R Validation Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
RCM	10.3	4.9	54.7	51.9	9.4	5.9	46.2	42.6
RCM+KED w/o speaker	9.2	6.5	55.9	54.0	8.7	7.1	48.5	43.3
RCM+KED with speaker	10.2	**3.8**	**64.1**	**61.1**	10.8	**5.4**	**49.7**	**45.2**

Table 5: Ablation study of speaker on **R2R** dataset.

Figure 3: visualization results of our model and the model w/o regularization objectives.

numbers generated by the baseline model have large distortion.

5 Conclusion

In this paper, we propose Knowledge-driven Environmental Dreamer (KED) to generate unseen scenes for agents to learn. KED is required to ensure texture consistency and structure consistency in generating new scenes. We propose three novel reconstruction objectives that leverage knowledge from pertained CLIP model and the "speaker" model to regularize the optimization of KED. Our experimental results reveal that the augmentation data generated by KED is able to significantly improve the performance of the navigation agent. Both quantitatively and qualitatively analysis infers that the knowledge-driven environmental dreamer is able to generate high-quality augmentation data.

Ethical Statement

Positive Impacts. Our work proposes a method to improve navigation performance, which contributes to real-world robotic applications. Household robots and rescue robots are required to navigate following a natural language instruction in an unseen house. Our work enables robots to complete navigation tasks accurately and efficiently without high-precision maps or hard-coded rules. Our work has a wide range of industrial applications, which could make today's robots more intelligent.

Risks and Ethical Concerns. In this paper, we use an openly published dataset, named Matterport3D [Chang *et al.*, 2017] for our research. Collecting 3D house scenes require consent from house owners to avoid violation of privacy. There is a local government "general data protection regulation" that a researcher needs to obtain prior ethical committee approval before conducting the investigation. As the end-to-end framework is vulnerable to abuse, security protection mechanisms should be incorporated to protect its embedded algorithm from hacking. If multiple languages are to be incorporated, then additional unified textual information protection techniques should be developed to alleviate natural language processing risk.

Acknowledgements

This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0109700, Nansha Key RD Program under Grant No.2022ZD014, Guangdong Outstanding Youth Fund (Grant No. 2021B1515020061) and the Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant No. 22lgqb38.

References

[Ammirato *et al.*, 2017] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C. Berg. A dataset for developing and benchmarking active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1378–1385, 2017.

[Anderson *et al.*, 2018a] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[Anderson *et al.*, 2018b] 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 *CVPR*, 2018.

[Bojarski *et al.*, 2016] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

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

[Chen *et al.*, 2017] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906, 2017.

[Chen *et al.*, 2021] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847, 2021.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Fried *et al.*, 2018] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, 2018.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[Hao *et al.*, 2020] 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 *CVPR*, 2020.

[Hertzmann *et al.*, 2001] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, New York, NY, USA, 2001. ACM Press.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

[Hong *et al.*, 2020] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, 2020.

[Hong *et al.*, 2021] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez Opazo, and Stephen Gould. VLN BERT: A recurrent vision-and-language BERT for navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1643–1653. Computer Vision Foundation / IEEE, 2021.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[Karras *et al.*, 2021] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[Koh *et al.*, 2021] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021.

[Kolve *et al.*, 2017] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[Li *et al.*, 2019] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *EMNLP-IJCNLP*, 2019.

[Liu *et al.*, 2021] Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, Zongyuan Ge, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1644–1654, 2021.

[Lu *et al.*, 2019] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[Majumdar *et al.*, 2020] 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. *arXiv preprint arXiv:2004.14973*, 2020.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

[Park *et al.*, 2020] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

[Qiao *et al.*, 2022] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427, 2022.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[Savva *et al.*, 2019] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *CVPR*, 2019.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tan *et al.*, 2019] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *ACL*, 2019.

[Tenenbaum and Freeman, 2000] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[Wang *et al.*, 2019] 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 *CVPR*, 2019.

[Wu *et al.*, 2018] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. In *ICLR*, 2018.

[Xia *et al.*, 2018] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018.

[Xia *et al.*, 2020] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020.

[Zhao *et al.*, 2021] Yizhou Zhao, Kaixiang Lin, Zhiwei Jia, Qiaozi Gao, Govind Thattai, Jesse Thomason, and Gaurav S Sukhatme. Luminous: Indoor scene generation for embodied ai challenges. *arXiv preprint arXiv:2111.05527*, 2021.

[Zhu *et al.*, 2020a] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020.

[Zhu *et al.*, 2020b] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.

[Zhu *et al.*, 2020c] Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. Multimodal text style transfer for outdoor vision-and-language navigation, 2020.

[Zhu *et al.*, 2021] Fengda Zhu, Yi Zhu, Vincent Lee, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021.