ENVEDIT: Environment Editing for Vision-and-Language Navigation

Jialu Li Hao Tan Mohit Bansal
UNC Chapel Hill
{jialuli, airsplay, mbansal}@cs.unc.edu

Abstract

In Vision-and-Language Navigation (VLN), an agent needs to navigate through the environment based on natural language instructions. Due to limited available data for agent training and finite diversity in navigation environments, it is challenging for the agent to generalize to new, unseen environments. To address this problem, we propose ENVEDIT*, a data augmentation method that creates new environments by editing existing environments, which are used to train a more generalizable agent. Our augmented environments can differ from the seen environments in three diverse aspects: style, object appearance, and object classes. Training on these edit-augmented environments prevents the agent from overfitting to existing environments and helps generalize better to new, unseen environments. Empirically, on both the Room-to-Room and the multi-lingual Room-Across-Room datasets, we show that our proposed* ENVEDIT *method gets significant improvements in all metrics on both pre-trained and non-pre-trained VLN agents, and achieves the new state-of-the-art on the test leaderboard. We further ensemble the VLN agents augmented on different edited environments and show that these edit methods are complementary.*[1]

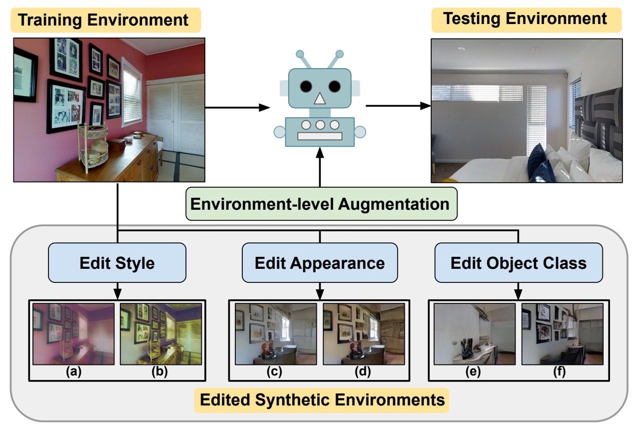

Figure 1. We create synthetic environments through editing the style (a, b), object appearance (c, d), and object classes (e, f) of the training environment. Our synthetic environments serve as environment-level data augmentation during training and help the agent's generalization to unseen testing environment.

1. Introduction

The Vision-and-Language Navigation (VLN) task requires an agent to navigate through the environment based on natural language instructions. Existing Vision-and-Language Navigation datasets are usually small in scale and contain a limited number of environments due to the difficulty of such data collection. However, navigation environments might differ greatly from each other. For example, indoor navigation environments might differ in the style of the room, the layout of the furniture, and the structure of the entire house. This makes it difficult for the agent to generalize to previously unseen environments. Previous works [14,21,23,28,40,60,66] have seen that agents perform substantially worse in unseen environments, and many thoughtful methods [18,22,37,41,56,61,62] have been proposed to solve this generalization problem. One line of the previous work focuses on augmenting the environments to mitigate the environment bias. For example, [56] proposes to drop out environment-level features during training. However, this feature-dropping approach lacks the interpretability of the actually modified environments that the agent learns from to gain better generalizability. [37] takes one step further in creating in-domain augmentation data by mixing up existing training environments, which effectively reduces generalization error of VLN agents. However, these mixed-up environments did not bring unseen changes or modifications to existing environments, and hence did not break the limitations of existing seen environments, which restricts agent's generalizability to unseen environments. Thus, in this paper, we propose to create new environments that differ from the original environments in style, appearance, and objects with style transfer and image synthesis approaches.

Another line of work tries to address environment bias by pre-training from large image-text datasets [18,41], which equips the agent with diverse visual knowledge. Although

[1]Code and data are available at https://github.com/jialuli-luka/EnvEdit.

promising performance was achieved, even pre-training data in [18], which is an indoor room environment with captions collected from AirBnB, still differs from the VLN task in two ways. In Vision-and-Language Navigation, the agent perceives a panoramic view and receives human-written language instructions, where in [18], the panoramic view is the concatenation of images with similar semantics, and the instruction is a template-based mixing of image descriptions. This leads to a domain shift in pre-training data and might not adapt well to the VLN task. Considering the large amount of pre-training data used, the performance gain on the Vision-and-Language Navigation task is still limited.

To address these challenges, in this work, we propose ENVEDIT: **Env**ironment **Edit**ing for Vision-and-Language Navigation. Our approach consists of three stages. In the first stage, we create new environments that maintain most of the semantic information of the original environments while changing the style, appearance, and object classes of the original environment. This constraint enables us to directly adopt the original human-annotated language instructions for the new environments, and avoid generating low-quality synthetic instructions [63]. As illustrated in Figure 1, our generated synthetic environments are mostly consistent with the original environments in semantics, but differ greatly in other aspects. For example, the overall style in Figure 1 (a, b) and object appearance in Figure 1 (c, d) are different, but the semantics of the synthetic environments mostly match the original environments. Meanwhile, our synthetic environments can also moderately differ from the original environments in object semantics (e.g., Figure 1 (e) removes the pictures from the wall). Learning from these synthetic environments could enable the agent to better understand visual semantics and be more robust to appearance changes of objects in different environments. Specifically, we adopt methods from style transfer [26] and image synthesis [49] to create new environments. In style transfer, the newly transferred environment is created with style embedding sampled from the learned embedding distribution of artistic paintings. In image synthesis, we generate new environments based on semantic segmentation of the original environments, which change the appearance of the objects. We further moderately edit the environment semantics and change objects (e.g., remove a lamp from the environment) by randomly masking some semantic classes in the semantic segmentation. In the second stage, the agent learns to navigate given natural language instructions from both the original environment and our aforementioned augmented environments. In the last stage, we follow the existing instruction-level data augmentation setup in [14, 56], which uses a speaker to generate new instructions for unannotated paths to fine-tune the agent. But different from [56], our speaker is aware of styles and can generate different instructions given the style of the environment.

We conduct experiments on both Room-to-Room (R2R) dataset [2] and the multi-lingual Room-Across-Room (RxR) dataset [30]. Empirical results show that our proposed ENVEDIT outperforms all other non-pre-training methods by 1.6% in success rate (SR) and 1.4% in success rate weighted by path length (SPL) on R2R test leaderboard, and 5.3% in normalized Dynamic Time Warping (nDTW) and 8.0% in success rate weighted by normalized Dynamic Time Warping (sDTW) on RxR test leaderboard. We further show that our proposed approach is beneficial to SotA pre-trained agents. Our ENVEDIT improves the performance by 3.2% in SR and 3.9% in SPL on R2R test leaderboard, and 4.7% in nDTW and 6.6% in sDTW on RxR test leaderboard, achieving the new state-of-the-art for both datasets. Lastly, we ensemble the VLN agents augmented on different edited environments and show that these editing methods are complementary to each other.

2. Related Work

Vision-and-Language Navigation. A lot of task setups, datasets, and simulators have been proposed for Vision-and-Language Navigation (VLN) [2,4,7,19,27,42,44,48,55,57]. In this paper, we focus on the Room-to-Room dataset [2] and Room-Across-Room dataset [30], which have human-annotated instructions in different languages and simulated environments captured in Matterport3D [6]. To solve this challenging task, a base agent contains cross-modality attention modules for cross-modal alignment between language and visual environment, LSTM [20] and transformer [58] based network to model context history and decode the sequence of navigation actions, and uses a mixture of reinforcement learning and imitation learning to train the agent [31, 34, 40, 56, 60, 65, 67]. In this paper, we build our methods on the strong baseline model EnvDrop [56] and further show our methods' compatibility to SotA pre-trained models [9, 22].

Mitigating Environment Bias. Generalization to unseen environments is a key challenge in Vision-and-Language Navigation, especially for real-world environments. Many works have been proposed to mitigate the environment bias and enhance the performance in unseen environments [18, 22, 37, 41, 56, 61, 62]. One line of work focuses on feature-level engineering [56, 61]. However, these methods lack interpretability of what new environments the agent actually perceives and what semantic information across environments is learned by the model. Another line of work focuses on pre-training on large amounts of image-text pairs from other resources (e.g., web, image-caption datasets) or adopting pre-trained weights from SotA Vision-and-Language transformers [35, 39] to inject common sense visual and text knowledge into the model for better generalizability to unseen environments [9, 18, 22, 41, 46]. Though

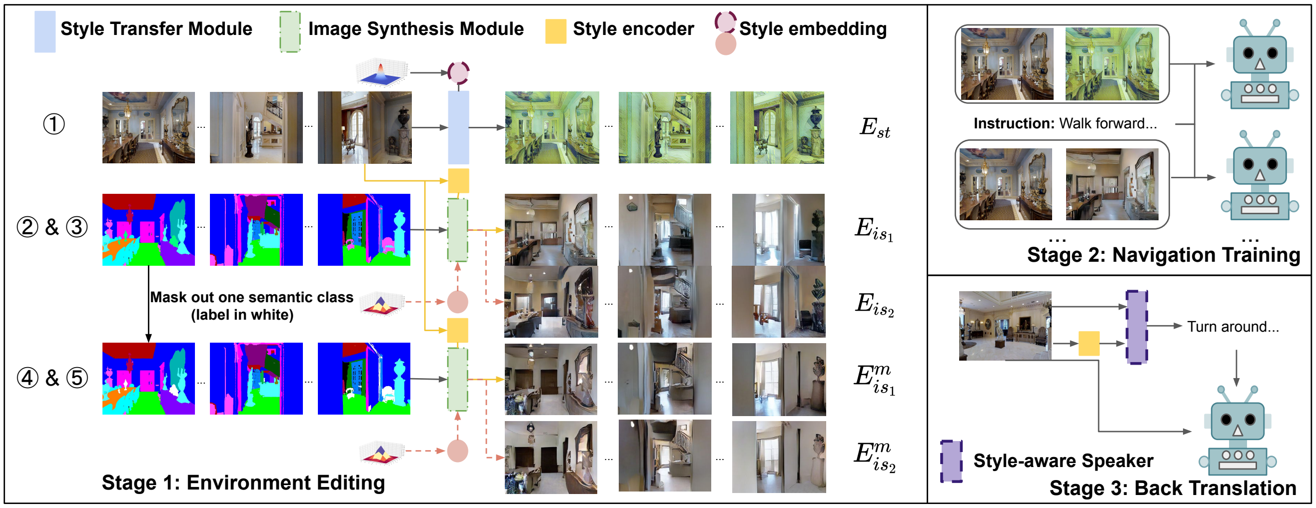

Figure 2. Overview of our ENVEDIT. In the first stage, the agent edits the original environment in five ways with style transfer and image synthesis approach (Sec. 4). Then, the agent is trained on Vision-and-Language Navigation task with both the original environments and the created environments (Sec. 3.2). Lastly, a style-aware speaker is utilized to generate synthetic instructions for unannotated paths for back translation (Sec. 3.2).

the pre-training methods show promising performance, the pre-training data still differs from the panorama observations and human-annotated instructions in Vision-and-Language Navigation. To address the domain-shift between pre-training data and VLN data, we propose to use style transfer and image synthesis methods to augment the existing VLN data with new environments.

Data Augmentation in Vision-and-Language Navigation. Data collection for Vision-and-Language Navigation task is resource consuming. Previous work in data augmentation mainly focuses on augmenting the instructions. [14, 56] propose to train a speaker that generates instructions given unannotated paths from the seen environments, and [68] proposes to transfer the style of text for instruction augmentation. However, how to augment the training environment for better generalization is still underexplored. [37] proposes a useful environment-mixing method that creates new paths by mixing sub-paths from different environments. However, their approach is still limited by the existing seen environments, since they only concatenate existing environments and do not create environments that are new in style and unseen to the agent. [29] first tries to synthesize existing environments and predict future scenes for Vision-and-Language Navigation. In contrast to them, we propose ENVEDIT that creates new environments for data augmentation and agents' generalization via style transfer and image synthesis methods.

Data Augmentation in Computer Vision. Data augmentation is a widely applied technique in the field of Computer Vision. Traditional data augmentation methods include random cropping, resizing, scaling, rotating, noise injection, image mixing, etc. [25, 32, 45]. With advances in deep neural networks, using GAN [17] based methods for data augmentation becomes popular [5, 15, 26, 50, 53, 64, 69]. Following this trend, we use style transfer [16] and image synthesis [49] to create new environments for data augmentation.

3. Method Overview

3.1. Problem Setup

Vision-and-Language Navigation (VLN) requires an agent to navigate through the environment based on natural language instructions. Formally, given a natural language instruction I, at each time step t, the agent perceives a panoramic view P_t of the current location, and needs to pick the next viewpoint from a set of K navigable locations $\{g_{t,k}\}_{k=1}^{K}$. Specifically, the panoramic view P_t is discretized into 36 single views $\{p_{t,i}\}_{i=1}^{36}$. Each view representation $f_{t,i}$ is the concatenation of its visual representation $v_{t,i}$ encoded by the pre-trained vision model and its orientation feature $o_{t,i} = (\cos\theta_{t,i}, \sin\theta_{t,i}, \cos\phi_{t,i}, \sin\phi_{t,i})$ that encodes heading $\theta_{t,i}$ and elevation $\phi_{t,i}$ information. The navigable location is represented as the visual feature of one specific view that is closest to the navigable direction from 36 discretized views and its orientation feature. The agent will predict a "STOP" action when the navigation ends.

3.2. Training Procedures

The overview of our **Env**ironment **Edit**ing (ENVEDIT) approach is shown in Figure 2. It contains three stages. We describe the three stages briefly in this section.

Environment Creation. In the first stage, we create multiple environments that differ from the original environments in style, appearance, and object classes (described in Sec 4.1 and Sec 4.2). We adopt the off-the-shelf models from [26] for style transfer. For image synthesis, we train the image generator and style encoder on all the seen environments in Room-to-Room dataset [2].

Vision-and-Language Navigation Training. In the second stage, the agent is trained on both the original environments and the new environments on the Vision-and-Language Navigation task. Specifically, in a batch of N instruction-path pairs, half of the pairs will observe the original environments, and the other half will perceive the edited environments. This prevents the agent from overfitting to the original environments. A mixture of imitation learning and reinforcement learning is adopted as in [56].

Back Translation. In the third stage, we follow [56] to do back translation, which generates synthetic instructions for unannotated paths from seen environments with a speaker. The agent is trained on both the original and the newly generated instruction-path pairs. The speaker used in [56] consists a two-layer bi-directional LSTM [20] that encodes route information $\{p_i\}_{i=1}^{L}$ and context information $\{c_i\}_{i=1}^{L}$ hierarchically, and a traditional LSTM based decoder with attention over encoded context information to generate synthetic instructions. To better serve the environment creation purpose, we enhance this speaker by further incorporating style information of the route. Specifically, we initialize the speaker decoder with the style embedding of the start viewpoint on the route:

$$x_0 = \text{LSTM}(w_0, (h_{style}, c_{style})) \tag{1}$$

$$(h_{style}, c_{style}) = \text{FCLayer}(s_0) \tag{2}$$

$$s_0 = \frac{1}{36}\sum_{k=1}^{36} \text{StyleEncoder}(o_{0,k}) \tag{3}$$

where $o_{0,k}$ is the discretized view for the start location and w_0 is the start token for instruction generation. x_0 is further attended with context information $\{c_i\}_{i=1}^{L}$ to predict the next word in the instruction.

4. Environment Editing

In this section, we describe the environment editing methods that we use to create new environments. We focus on editing three components (S, A, O) of an environment E, where S is the style of the environment, A is the object appearance, and O is the class of objects (indicated by the semantic segmentation mask of the environment). In Sec 4.1, we present the style transfer we use to edit the style S of an environment. In Sec 4.2, we use image synthesis to edit the object appearance A of an environment, while also providing the option to edit the style S and the objects O. An example of different kinds of created environments is shown in Figure 2 Stage 1.

4.1. Style Transfer

Previous environment-level data augmentation methods in Vision-and-Language Navigation mainly focus on feature augmentation (i.e., adding random noise directly to the visual representation encoded with pre-trained vision models) [56] and environment mixing augmentation (i.e., mixing up paths from two training environments) [37]. Though both methods are useful and achieve promising results, feature augmentation has the issue of hard to interpret, and environment mixing augmentation does not address the limitations of existing environments and sometimes the mixed scenes are unrealistic (e.g., navigating from a modern living house to a museum-style room). To address these issues, we propose to create new environments that are semantically consistent with the original environments but different in style. Our created new environments could potentially mimic the unseen environments, and are more realistic compared with [37]. The main advantage of maintaining the semantics of the original environments is that the original human-annotated language instructions can be directly adapted to the new environments with high correspondence. This eliminates the need to generate synthetic instructions [14,56] for new environments, which has shown to be much worse than human annotations [63]. The new environment E_{st} (st – **S**tyle **T**ransfer) we create is (S_{st}, A_o, O_o), which differs from the original environment E_o in style of the environment. Specifically, we follow the approach [26] that is both computationally efficient and has high quality output.

Style Transfer Model Architecture. The architecture of the style transfer approach we use is shown in Figure 2 Stage 1 row ①. The content image is encoded by a CNN based architecture with residual connections. The style embedding is sampled from a multivariate normal distribution, whose mean and covariance is from the distribution of the style embeddings from the Painter By Numbers (PBN) dataset[2]. During decoding, the sampled style embedding is incorporated by the conditional instance normalization [12]:

$$x_{out} = \gamma_{style}\left(\frac{x_{in} - \mu_{in}}{\sigma_{in}}\right) + \beta_{style} \tag{4}$$

where γ_{style} and β_{style} are computed by passing the sampled style embedding through two separate fully-connected layers, and μ_{in} and σ_{in} are the mean and standard deviation of the encoded content image x_{in} respectively.

[2]https://www.kaggle.com/c/painter-by-numbers

Fixed style for discretized views. At each time step, the agent perceives a panoramic view P_t of the current location, which is discretized into 36 single views $\{p_{t,i}\}_{i=1}^{36}$. The 36 single views are correlated to each other, and there exists overlaps between adjacent views. Thus, to keep the visual observation consistent in style in a panoramic view at one time step, we sample the same style embedding from the multivariate normal distribution for all the 36 discretized views. We show in Sec 6.3 that this setup is crucial for creating effective edited environments for the agent.

4.2. Image Synthesis

The style transfer approach creates environments that maintain the semantics of the original environments and only change the style. Nevertheless, the appearance and texture of objects in the environment remain the same. Thus, we explore one step further by creating new environments that are semantically similar to the original environments but different in both style and appearance of the objects with the image synthesis approach. We explore a specific form of conditional image synthesis, which generates a new photorealistic image conditioned on a semantic segmentation mask. In this setup, the semantics of the new environments E_{is} are constrained by the semantic segmentation of the original environments, while the shape and appearance A of the objects can be diversely generated by the model. We further explore generating environments that have different objects O by changing one of the semantic classes in the semantic segmentation. In both cases, the synthetic environments have high correspondence with the original instructions, since the semantics remain unchanged or only slightly different from the original environments. We adopt the approach from [49] for semantic image synthesis.

Model Architecture. The image generator is a GAN based conditional image synthesis model as in [49]. Specifically, the model contains several ResNet blocks with upsampling layers. SPADE blocks [49] are used to learn the parameters for normalization layers and are conditioned on semantic segmentation mask information. The model is trained on GAN hinge loss [36] and feature matching loss [59]. Given the semantic segmentation mask, we could control the style of the synthesis image by using different style embeddings as input to the generator. Following [49], we learn an encoder that maps a style image to a style embedding by adding a KL-Divergence loss during training.

Editing Appearance. With the image generator and style encoder, we create two kinds of environments that edit the appearance of the original environments (shown in Figure 2 Stage 1 row ② and ③). The first kind of environment is E_{is_1} (is – **I**mage **S**ynthesis) with components (S_o, A_{is_1}, O_o), which differs from the original environments only in object appearance. We create this kind of environment by using the views in the original environments as the style image to maintain the style of the original environments. The second kind of environment is E_{is_2} with components $(S_{is_2}, A_{is_2}, O_o)$, created by manually setting a fixed style embedding (e.g., all-zero embedding) for the generator. This new environment differs from the original environments E_o in both style and object appearance. With these two kinds of environments, we are able to explore the impact of the style S and object appearance A separately.

Editing Objects. After creating environments with different styles S and object appearances A, we take one step further to remove and change some of the objects in the original environments (shown in Figure 2 Stage 1 row ④ and ⑤). Though there exists many works in (text-guided) image manipulation [3, 8, 10, 11, 13, 33, 38, 43, 47, 54], for simplicity, we change the objects by modifying the semantic segmentation of the original environments. Specifically, suppose that the original semantic segmentation contains C classes, we add a "mask" class and use it as the $C + 1$ class. During training, we randomly pick one class from the C classes and set it to be the "mask" class. In this case, the model could generate random masks for the "mask" class. We create new environments $E^m_{is_1}(S_o, A^m_{is_1}, O^m_{is_1})$ and $E^m_{is_2}(S^m_{is_2}, A^m_{is_2}, O^m_{is_2})$ (E^m_{is} – **I**mage **S**ynthesis with **M**asks) by randomly masking out objects from the original environment E_o. $E^m_{is_1}$ and $E^m_{is_2}$ differ in that $E^m_{is_1}$ maintains the style of the original environment, while $E^m_{is_2}$ changes all three components of the original environment E_o. The style changes are controlled by the style embedding from the style encoder.

5. Experimental Setup

5.1. Datasets

We evaluate our agent on the Room-to-Room (R2R) dataset [2] and the Room-Across-Room (RxR) dataset [30]. R2R dataset contains English instructions and RxR dataset contains instructions in English, Hindi, and Telugu. Both datasets are split into a training set, seen and unseen validation sets, and a test set. The environments in the unseen validation set and test set do not appear in the training set.

5.2. Evaluation Metrics

We evaluate our model with six metrics: (1) Success Rate (SR). (2) Success Rate weighted by Path Length (SPL) [1]. (3) Trajectory Length (TL). (4) Navigation Error (NE). (5) normalized Dynamic Time Warping (nDTW) [24]. (6) success rate weighted by normalized Dynamic Time Warping (sDTW) [24]. SR, SPL are the main metrics for evalu-

	Environment Components			ViT-B/32				ViT-B/16			
Models	Style	Appearance	Object	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
EnvDrop* [52]	✗	✗	✗	14.339	5.214	51.3	45.8	15.861	4.734	55.1	48.8
E_{st}	✓	✗	✗	14.738	**4.631**	**56.5**	**50.7**	16.585	4.690	**58.2**	**51.5**
E_{is_1}	✗	✓	✗	15.871	4.766	56.2	49.8	17.690	4.759	56.4	48.9
E_{is_2}	✓	✓	✗	15.427	5.049	54.2	48.5	15.273	4.767	56.2	49.6
$E^m_{is_1}$	✗	✓	✓	15.788	4.966	54.2	49.7	14.464	4.666	57.3	51.1
$E^m_{is_2}$	✓	✓	✓	17.906	4.979	54.2	47.6	14.204	**4.607**	56.1	50.8

Table 1. Performance of training the agent with one kind of our edited environments. Results are on R2R val-unseen set. ViT-B/32(16) indicate image features extracted with different CLIP-ViT models [51]. "*" indicates reproduced results. ✓ indicates the environment component of the new environment is different from the original environment, while ✗ indicates the same.

ation on R2R dataset, and nDTW, sDTW are the main metrics for RxR dataset. Details can be found in Appendix.

6. Results and Analysis

In this section, we first compare the performance of training on different environments that we created in Sec. 6.1. Then, we show that our method could generalize to pre-trained navigation agents in Sec. 6.2. We further show the importance of using a fixed style for discretized views and our style-aware speaker through ablations in Sec. 6.3. Moreover, we demonstrate that our created environments are complementary to each other in Sec. 6.4. Lastly, we show our model's performance on the test leaderboards of both Room-to-Room dataset and Room-Across-Room dataset in Sec. 6.5. We demonstrate some qualitative examples of our edited environments in Sec. 6.6.

6.1. Results for Environment Editing Methods

In the environment creation stage, we create five kinds of environments that differ in style, appearance and objects $E_{st}, E_{is_1}, E_{is_2}, E^m_{is_1}, E^m_{is_2}$ with editing methods as described in Sec 4.1 and Sec 4.2. We show the performance of training with the original environments and one of the new environments on R2R dataset in Table 1. Back translation is not applied in these experiments, and could be found in Appendix. We can see that training with any of the newly created environments can outperform the baseline model by a large margin on the validation unseen set. Specifically, the E_{st} environment, which differs from the original environment in style only, achieves the best performance, improving the baseline trained on ViT-B-32 features by 5.2% in SR and 4.9% in SPL, and a stronger baseline trained on ViT-B-16 features by 3.1% in SR and 2.7% in SPL. This demonstrates that augmenting the training environment with synthetic new environments helps generalization to unseen data, regardless of the visual features.

Overall, comparing the environments created with style transfer approach E_{st} and the environments created with image synthesis approaches $\{E_{is_1}, E_{is_2}, E^m_{is_1}, E^m_{is_2}\}$, E_{st} brings slightly higher improvement in both SR and SPL. We attribute this to the higher environment creation quality of the style transfer approach compared with the image synthesis approach. Comparing E_{is_1} and E_{is_2}, it shows that keeping the style unchanged while modifying the appearance of the objects improves the SR by 2.0% for model trained with ViT-B/32 features and 0.2% for model trained with ViT-B/16 features. This is because that the new environments that maintain the style will have a higher correspondence with the original instructions, while also being different enough. Given that we do not generate synthetic instructions for new environments (due to the low quality of synthetic instruction [63]), it is important to find a balance between the matches to the original instructions and the diversity of the new environments. Similar results are observed for $E^m_{is_1}$, $E^m_{is_2}$.

Furthermore, comparing $\{E_{is_1}, E^m_{is_1}\}$, we observe that features that learned from smaller patches (ViT-B/16) could benefit from the slight removal or change of objects in the environments. Similar performance improvement is observed for pre-trained VLN agent with ViT-B/16 features (discussed in Sec. 6.2).

Lastly, we observe that with different visual backbones (ViT-B/32 and ViT-B/16), and different VLN base models (discussed in Sec. 6.2), the improvement brought by different synthetic environments are inconsistent. For example, with the same base agent EnvDrop, training on E_{is_1} works better than training on $E^m_{is_1}$ for ViT-B/32 features, and not for ViT-B/16 features. We attribute this to features extracted with different visual backbones generalize to unseen environments differently. Detailed analysis can be found in the Appendix. Considering both simplicity and performance across different visual backbone and VLN base models, we recommend using E_{st} as a start point in future research.

6.2. Performance on Pre-trained VLN Agents

In this section, we show that our EnvEdit is complementary to the VLN pre-training methods. We enhance the SotA pre-traind VLN model HAMT [9] with our methods and illustrate the improvements on R2R dataset.

The model architecture of [9] is based on transformer. The image feature used in this work is extracted with CLIP

Models	TL	NE↓	SR↑	SPL↑
HAMT [9]	-	-	65.7	60.9
E_{st}-16	11.78	3.42	67.3	62.6
E_{is_1}-16	11.23	3.52	66.8	62.1
$E^m_{is_1}$-16	12.13	**3.22**	**67.9**	**62.9**

Table 2. Performance of applying our proposed method to SotA VLN agents on R2R validation unseen set.

Models	TL	NE↓	SR↑	SPL↑
EnvDrop-16* [52]	15.86	4.73	55.1	48.8
E_{st}-16	16.59	**4.69**	**58.2**	**51.5**
E_{st}-16 w/o fixed views	16.79	4.70	55.9	48.8
E_{st}-16 w/ fixed env	14.36	4.70	55.8	49.5
EnvDrop-32* [52]	14.34	5.21	51.3	45.8
E_{st}-32	14.74	**4.63**	**56.5**	**50.7**
E_{st}-32 w/o fixed views	14.50	4.88	54.7	48.8
E_{st}-32 w/ fixed env	17.39	4.87	55.0	48.5

Table 3. Ablation results on R2R val-unseen set illustrating the benefit of using the fixed style for a panorama. "-16" and "-32" indicate image features extracted with ViT-B/16(32). "*" indicates reproduced results.

ViT-B/16 without the last linear representation layer. We follow their work to extract the visual features for our created environments, and directly fine-tune their released pre-trained models with ENVEDIT. As shown in Table 2, augmenting the original environment with $E^m_{is_1}$ could improve the baseline by 2.2% in SR and 1.9% in SPL. Augmenting with the other two environments could also improve the baseline by more than 1.5% in both SR and SPL. This demonstrates the effectiveness of adapting our method to strong SotA VLN models.

6.3. Method Ablations

In this section, we show two ablations for our proposed method. We first show that using a fixed style for all 36 discretized views of a panorama is essential for creating new environments for the agent. Then, we show that our style-aware speaker achieves better performance when used in back translation compared with baseline speaker.

Fixed style for discretized views. As shown in Table 3, when the style is different inside a panorama ("E_{st}-16" vs. "E_{st}-16 w/o fixed views"), the performance drops by 2.3% in SR and 1.7% in SPL. This indicates that using a fixed style for 36 discretized views of a panorama is essential for the performance improvement, since it provides consistent visual semantics. Furthermore, we show that keeping a fixed style for the whole environment ("E_{st}-16 w/ fixed env") will decrease the improvements by 2.4% in SR and 2.0% in SPL. Similar results are observed for ViT-B/32 features. This indicates that using a fixed style at each viewpoint has a better balance between consistency in observation and variance in style.

Models	TL	NE↓	SR↑	SPL↑
E_{st}-32 + BT	17.777	4.504	59.0	51.8
E_{st}-32 + BTS	15.912	**4.335**	**60.2**	**53.8**
$E^m_{is_1}$-16 + BT	16.752	4.316	60.2	53.4
$E^m_{is_1}$-16 + BTS	17.989	**4.232**	**60.8**	**54.2**

Table 4. Ablation results on R2R val-unseen set showing the improvement of our style-aware speaker. '+BT" indicates back translation with the baseline speaker, and "+BTS" indicates using style-aware speaker in back translation.

Models	TL	NE↓	SR↑	SPL↑
E_{st}-ED	16.59	4.69	58.2	51.5
E_{st}+E_{is_1}+$E^m_{is_1}$-ED	15.60	**4.52**	**58.8**	**52.7**
$E^m_{is_1}$-H	12.13	3.22	67.9	62.9
E_{st}+E_{is_1}+$E^m_{is_1}$-H	11.13	**3.24**	**68.9**	**64.4**

Table 5. Performance of ensembling VLN agents trained on different environments. "ED" and "H" indicates using EnvDrop and HAMT as the base navigation agents respectively.

Style-aware Speaker. As shown in Table 4, our style-aware speaker improves the performance in SR and SPL by around 1% for both features (ViT-B/32 and ViT-B/16). This implies that explicitly incorporating environment style helps generate synthetic instructions that match with the environments better. Besides, we show that our style-aware speaker can improve the overall performance for different kinds of created environments (i.e., E_{st}, $E^m_{is_1}$).

6.4. Combining Multiple Environments

In this section, we discuss our initial exploration for combining multiple environments, where we use the traditional ensemble method to boost overall performance. Specifically, the agent makes its decision based on the average logits predicted by all the ensembled models. As shown in Table 5, for agents that use EnvDrop as the base agent, simply ensembling the VLN agents trained on three edited environments (E_{st}, E_{is_1}, $E^m_{is_1}$) could slightly improve the overall performance by 0.6% in SR and 1.2% in SPL compared with augmenting only with E_{st}. Similar improvement is observed when using HAMT as the base agent. We further explore combining multiple environments during training using adaptive curriculum learning in the Appendix.

6.5. Test Set Results

We show our method's performance on both the Room-to-Room (R2R) and the multi-lingual Room-Across-Room (RxR) leaderboards. All our agents are tested under the single-run setting, where the agent only navigates once and does not pre-explore the test environment.

On R2R dataset, we first compare our ENVEDIT with non-pre-training methods. Specifically, we apply our ENVEDIT to EnvDrop-CLIP [52], and train the model on $E^m_{is_1}$ with ViT-B/16 features. As shown in Table 6,

Models	Validation Seen				Validation Unseen				Test Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
↻BERT♠ [22]	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
EnvDrop-CLIP [52]	-	-	-	-	-	-	59.2	52.9	-	-	59	53
AirBERT♠ [18]	11.09	2.68	75	70	11.78	4.01	62	56	12.41	4.13	62	57
HAMT♠ [9]	11.15	2.51	76	72	11.46	**2.29**	66	61	12.27	3.93	65	60
REM♠ [37]	10.88	2.48	75.4	71.8	12.44	3.89	63.6	57.9	13.11	3.87	65.2	59.1
Ours	14.64	3.35	69.4	64.2	17.99	4.23	60.8	54.2	16.84	4.30	60.6	54.4
Ours♠	11.18	**2.32**	**76.9**	**73.9**	11.13	3.24	**68.9**	**64.4**	11.90	**3.59**	**68.2**	**63.9**

Table 6. Comparison of agent performance on R2R dataset under the single-run setting. ♠ indicates pre-trained VLN agents.

Models	Validation Seen			Validation Unseen			Test Unseen		
	SR↑	NDTW↑	SDTW↑	SR↑	NDTW↑	SDTW↑	SR↑	NDTW↑	SDTW↑
EnvDrop-CLIP [52]	-	-	-	42.6	55.7	-	38.3	51.1	32.4
HAMT♠ [9]	59.4	65.3	50.9	56.5	63.1	48.3	53.1	59.9	45.2
Ours	53.1	63.0	45.9	50.1	60.6	43.0	46.2	56.4	40.4
Ours♠	**67.2**	**71.1**	**58.5**	**62.8**	**68.5**	**54.6**	**60.4**	**64.6**	**51.8**

Table 7. Comparison of agent performance on RxR dataset under the single-run setting. ♠ indicates pre-trained VLN agents.

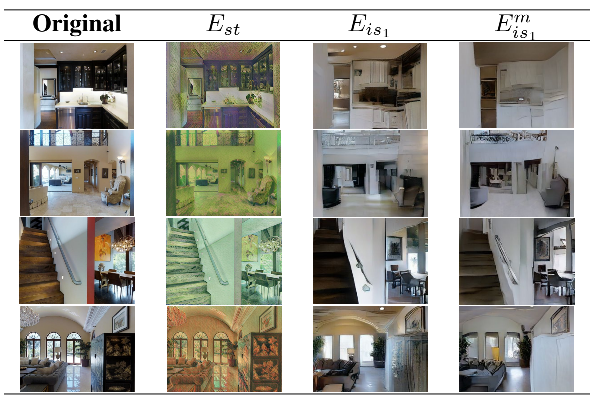

Table 8. Qualitative Examples of our edited environments.

it outperforms the previous best non-pre-training method ("EnvDrop-CLIP") by 1.6% in SR and 1.4% in SPL on test unseen set. We further adapt our ENVEDIT to pre-trained SotA model HAMT [9]. The model shown in Table 6 is an ensemble of models trained on E_{st}, E_{is_1} and $E^m_{is_1}$. Our ENVEDIT outperforms the HAMT by 3.2% in SR and 3.9% in SPL, and achieves the new SotA on the leaderboard.

On the multi-lingual RxR dataset, we first apply our ENVEDIT on non-pre-trained SotA model EnvDrop-CLIP [52], where we replace the LSTM based instruction encoder with multi-lingual BERT. We train the model with E_{st}, and utilize ViT-B/16 to extract visual features. As shown in Table 6, our ENVEDIT surpasses the previous best non-pre-training method ("EnvDrop-CLIP") by 5.3% in nDTW and 8.0% in sDTW. We further adapt our ENVEDIT to pre-trained SotA model HAMT [9]. For a fair comparison with HAMT, we use ViT-B/32 features and do not train the visual backbone end-to-end. Ensembling VLN agents trained on three edited environments (E_{st}, E_{is_1}, $E^m_{is_1}$) outperforms HAMT by 4.7% in nDTW and 6.6% in sDTW on the test leaderboard, achieving the new SotA for RxR dataset.

6.6. Qualitative Analysis for Edited Environments

We show some examples for our edited environments in Table 8. We could see that the environments generated with the style transfer approach (E_{st}) maintain the semantics of the original environments better with overall artistic style. The environments generated with the image synthesis approach (E_{is_1} and $E^m_{is_1}$) change object appearances and are more close to real environments. For example, in the last row of $E^m_{is_1}$, the cabinet is masked out during image generation, which brings more diversity in the environments.

7. Conclusion and Discussion

In this paper, we present ENVEDIT, which augments the Vision-and-Language Navigation training by editing existing environments. Our created environments differ from the original environments in the overall style, object appearances, and object classes, thus can mimic the unseen environments. Our experiments on both Room-to-Room and Room-Across-Room datasets show that training on the edited environments improves the performance in all evaluation metrics compared with both pre-training and non-pre-training methods, and achieves the new SotA on the test leaderboard. Furthermore, we ensemble the VLN agents trained on different edited environments and show that these environments are complementary to each other.

Acknowledgement

We thank the reviewers, Jaemin Cho, Yi-Lin Sung, Hyounghun Kim, Jie Lei, and Zhenlin Xu for their helpful comments. This work was supported by NSF-CAREER Award 1846185, ARO W911NF2110220, ONR N000141812871, DARPA KAIROS FA8750-19-2-1004, Google Focused Award. The views contained in this article are those of the authors and not of the funding agency.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 5

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 2, 4, 5

[3] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 5

[4] Matthew Berg, Deniz Bayazit, Rebecca Mathew, Ariel Rotter-Aboyoun, Ellie Pavlick, and Stefanie Tellex. Grounding language to landmarks in arbitrary outdoor environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 208–215. IEEE, 2020. 2

[5] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018. 3

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2

[7] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 2

[8] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8721–8729, 2018. 5

[9] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2, 6, 7, 8

[10] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention gan for interactive image editing. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4383–4391, 2020. 5

[11] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5706–5714, 2017. 5

[12] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *ICLR*, 2017. 4

[13] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10304–10312, 2019. 5

[14] 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 *Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 3, 4

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*, 2019. 3

[16] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. 3

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[18] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021. 1, 2, 8

[19] Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. Learning to follow directions in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11773–11781, 2020. 2

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 4

[21] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[22] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, June 2021. 1, 2, 8

[23] Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7404–7413, 2019. 1

[24] 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 (ViGIL) Workshop*, 2019. 5

[25] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 3

[26] Philip TG Jackson, Amir Atapour Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: data augmentation via style randomization. In *CVPR Workshops*, pages 83–92, 2019. 2, 3, 4

[27] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872, Florence, Italy, July 2019. Association for Computational Linguistics. 2

[28] Hyounghun Kim, Jialu Li, and Mohit Bansal. NDH-full: Learning and evaluating navigational agents on full-length dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6432–6442, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. 1

[29] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. *ICCV*, 2021. 3

[30] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020. 2, 5

[31] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Multimodal attention networks for low-level vision-and-language navigation. *Computer Vision and Image Understanding*, 2021. 2

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[33] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020. 5

[34] Jialu Li, Hao Tan, and Mohit Bansal. Improving cross-modal alignment in vision language navigation via syntactic information. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1050, Online, June 2021. Association for Computational Linguistics. 2

[35] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 2

[36] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 5

[37] 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. 1, 2, 3, 4, 8

[38] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe what to change: A text-guided unsupervised image-to-image translation approach. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1357–1365, 2020. 5

[39] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 13–23, 2019. 2

[40] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1, 2

[41] 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. 1, 2

[42] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2

[43] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 5

[44] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3D environments with visual goal prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. 2

[45] Francisco J Moreno-Barea, Fiammetta Strazzera, José M Jerez, Daniel Urda, and Leonardo Franco. Forward noise adjustment scheme for data augmentation. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pages 728–734. IEEE, 2018. 3

[46] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene-and object-aware transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[47] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 5

[48] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP), pages 684–695, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 2

[49] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2, 3, 5

[50] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 2017. 3

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[52] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *ICLR*, 2022. 6, 7, 8

[53] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *International workshop on simulation and synthesis in medical imaging*, pages 1–11. Springer, 2018. 3

[54] Seitaro Shinagawa, Koichiro Yoshino, Sakriani Sakti, Yu Suzuki, and Satoshi Nakamura. Interactive image manipulation with natural language instruction commands. *Advances in neural information processing systems, ViGIL workshop*, 2017. 5

[55] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 2

[56] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2610–2621, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 1, 2, 3, 4

[57] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 5

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 1, 2

[61] Xin Eric Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 413–430. Springer, 2020. 1, 2

[62] Yubo Zhang, Hao Tan, and Mohit Bansal. Diagnosing the environment bias in vision-and-language navigation. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 890–897. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. 1, 2

[63] Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alexander Ku, Jason Baldridge, and Eugene Ie. On the evaluation of vision-and-language navigation instructions. *EACL*, 2021. 2, 4, 6

[64] Xu Zheng, Tejo Chalasani, Koustav Ghosal, Sebastian Lutz, and Aljosa Smolic. Stada: Style transfer as data augmentation. *14th International Conference on Computer Vision Theory and Applications*, 2019. 3

[65] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021. 2

[66] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2020. 1

[67] 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. In *ACL*, pages 2539–2556, 2020. 2

[68] Wanrong Zhu, Xin Eric 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. *EACL*, 2021. 3

[69] Xinyue Zhu, Yifan Liu, Jiahong Li, Tao Wan, and Zengchang Qin. Emotion classification with data augmentation using generative adversarial networks. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 349–360. Springer, 2018. 3