Multimodal Text Style Transfer for Outdoor Vision-and-Language Navigation

Wanrong Zhu[†], **Xin Eric Wang**[‡], **Tsu-Jui Fu**[†], **An Yan**[§],
Pradyumna Narayana[*], **Kazoo Sone**[*], **Sugato Basu**[*], **William Yang Wang**[†]
[†]UC Santa Barbara, [‡]UC Santa Cruz, [§]UC San Diego, [*]Google
{wanrongzhu,tsu-juifu,william}@cs.ucsb.edu, xwang366@ucsc.edu,
ayan@eng.ucsd.edu, {pradyn,sone,sugato}@google.com

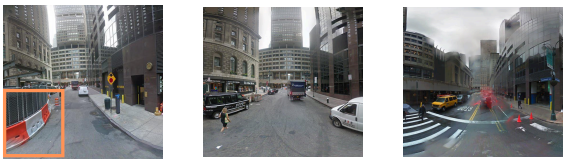
Abstract

One of the most challenging topics in Natural Language Processing (NLP) is visually-grounded language understanding and reasoning. Outdoor vision-and-language navigation (VLN) is such a task where an agent follows natural language instructions and navigates a real-life urban environment. Due to the lack of human-annotated instructions that illustrate intricate urban scenes, outdoor VLN remains a challenging task to solve. This paper introduces a Multimodal Text Style Transfer (MTST) learning approach and leverages external multimodal resources to mitigate data scarcity in outdoor navigation tasks. We first enrich the navigation data by transferring the style of the instructions generated by Google Maps API, then pre-train the navigator with the augmented external outdoor navigation dataset. Experimental results show that our MTST learning approach is model-agnostic, and our MTST approach significantly outperforms the baseline models on the outdoor VLN task, improving task completion rate by 8.7% relatively on the test set. [1]

1 Introduction

A key challenge for Artificial Intelligence research is to go beyond static observational data and consider more challenging settings that involve dynamic actions and incremental decision-making processes (Fenton et al., 2020). Outdoor vision-and-language navigation (VLN) is such a task, where an agent navigates in an urban environment by grounding natural language instructions in visual scenes, as illustrated in Fig. 1. To generate a series of correct actions, the navigation agent must comprehend the instructions and reason through the visual environment.

[1]Our code and dataset is released at https://github.com/VegB/VLN-Transformer.



Google Maps API Vanderbilt Ave **turns right** and becomes E 43rd St.

Speaker You ' ll have a red brick building with a red awning on your right . Go forward until you reach the next intersection , and turn right.

MTST model Turn right again and stop just past the orange and white construction barriers.

Figure 1: An outdoor VLN example with instructions generated by Google Maps API (ground truth), the Speaker model, and our MTST model. Tokens marked in red indicate incorrectly generated instructions, while the blue tokens suggest alignments with the ground truth. The orange bounding boxes show that the objects in the surrounding environment have been successfully injected into the style-modified instruction.

Different from indoor navigation (Anderson et al., 2018; Wang et al., 2018; Fried et al., 2018; Wang et al., 2019; Ma et al., 2019a; Tan et al., 2019; Ma et al., 2019b; Ke et al., 2019), the outdoor navigation task takes place in urban environments that contain diverse street views (Mirowski et al., 2018; Chen et al., 2019; Mehta et al., 2020). The vast urban area leads to a much larger space for an agent to explore and usually contains longer trajectories and a wider range of objects for visual grounding. This requires more informative instructions to address the complex navigation environment. However, it is expensive to collect human-annotated instructions that depict the complicated visual scenes to train a navigation agent. The issue of data scarcity limits the navigator's performance in the outdoor VLN task.

To deal with the data scarcity issue, Fried et al. (2018) proposes a Speaker model to generate additional training pairs. However, synthesizing instructions purely from visual signals is hard, especially for outdoor environments, due to visual complexity.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1207–1221
April 19 - 23, 2021.

On the other hand, template-based navigation instructions on the street view can be easily obtained via the Google Map API, which may serve as additional learning signals to boost outdoor navigation tasks. But instructions generated by Google Maps API mainly consist of street names and directions, while human-annotated instructions in the outdoor navigation task frequently refer to street-view objects in the panorama. The distinct instruction style hinders the full utilization of external resources.

Therefore, we present a novel Multimodal Text Style Transfer (MTST) learning approach to narrow the gap between template-based instructions in the external resources and the human-annotated instructions for the outdoor navigation task. It can infer style-modified instructions for trajectories in the external resources and thus mitigate the data scarcity issue. Our approach can inject more visual objects in the navigation environment to the instructions (Fig. 1), while providing direction guidance. The enriched object-related information can help the navigation agent learn the grounding between the visual environment and the instruction.

Moreover, different from previous LSTM-based navigation agents, we propose a new VLN Transformer to predict outdoor navigation actions. Experimental results show that utilizing external resources provided by Google Maps API during the pre-training process improves the navigation agent's performance on Touchdown, a dataset for outdoor VLN (Chen et al., 2019). In addition, pre-training with the style-modified instructions generated by our multimodal text style transfer model can further improve navigation performance and make the pre-training process more robust. In summary, the contribution of our work is four-fold:

- We present a new Multimodal Text Style Transfer learning approach to generate style-modified instructions for external resources and tackle the data scarcity issue in the outdoor VLN task.
- We provide the Manh-50 dataset with style-modified instructions as an auxiliary dataset for outdoor VLN training.
- We propose a novel VLN Transformer model as the navigation agent for outdoor VLN and validate its effectiveness.
- We improve the task completion rate by 8.7% relatively on the test set for the outdoor VLN task with the VLN Transformer model pre-trained on the external resources processed by our MTST approach.

2 Related Work

Vision-and-Language Navigation (VLN) is a task that requires an agent to achieve the final goal based on the given instructions in a 3D environment. Besides the generalizability problem studied by previous works (Wang et al., 2018, 2019; Tan et al., 2019; Zhang et al., 2020), the data scarcity problem is another critical issue for the VLN task, expecially in the outdoor environment(Chen et al., 2019; Mehta et al., 2020; Xiang et al., 2020). Fried et al. (2018) obtains a broad set of augmented training data for VLN by sampling trajectories in the navigation environment and using the Speaker model to back-translate their instructions. However, the Speaker model might cause the error propagation issue since it is not trained on large corpora to optimize generalization. While most existing works select navigation actions dynamically along the way in the unseen environment during testing, Majumdar et al. (2020) proposes to test in previously explored environments and convert the VLN task to a classification task over the possible paths. This approach performs well in the indoor setting, but is not suitable for outdoor VLN where the environment graph is different.

Multimodal Pre-training has attracted much attention to improving multimodal tasks performances. The models usually adopt the Transformer structure to encode the visual features and the textual features (Tan and Bansal, 2019; Lu et al., 2019; Chen et al., 2020; Sun et al., 2019; Li et al., 2019; Huang et al., 2020b; Luo et al., 2020; Li et al., 2020; Zheng et al., 2020; Wei et al., 2020; Tsai et al., 2019). During pre-training, these models use tasks such as masked language modeling, masked region modeling, image-text matching to learn the cross-modal encoding ability, which later benefits the multimodal downstream tasks. Majumdar et al. (2020) proposes to use image-text pairs from the web to pre-train VLN-BERT, a visiolinguistic transformer-based model similar to the model proposed by Lu et al. (2019).

A concurrent work by Hao et al. (2020) proposes to use Transformer for indoor VLN. Our VLN Transformer is different from their model in several key aspects: (1) The pre-training objectives are different: Hao et al. (2020) pre-trains the model

on the same dataset for training, while we create an augmented, stylized dataset for outdoor VLN using the proposed MTST method. (2) Benefiting from the effective external resource, a simple navigation loss is employed in our VLN Transformer, while they adopt the masked language modeling to better train their model. (3) Model-wise, instead of encoding the whole instruction into one feature, we use sentence-level encoding since Touchdown instructions are much longer than R2R instructions. (4) We encode the trajectory history, while their model encodes the panorama for the current step.

Unsupervised Text Style Transfer is an approach to mitigate the lack of parallel data for supervised training. One line of work encodes the text into a latent vector and manipulate the text representation in the latent space to transfer the style. Shen et al. (2017); Hu et al. (2017); Yang et al. (2018) use variational auto-encoder to encode the text, and use a discriminator to modify text style. John et al. (2019); Fu et al. (2018) rely on models with encoder-decoder structure to transfer the style. Another line of work enriches the training data by generating pseudo-parallel data via back-translation (Artetxe et al., 2018; Lample et al., 2018b,a; Zhang et al., 2018).

3 Methods

3.1 Task Definition

In the vision-and-language navigation task, the reasoning navigator is asked to find the correct path to reach the target location following the instructions (a set of sentences) $\mathcal{X} = \{s_1, s_2, \ldots, s_m\}$. The navigation procedure can be viewed as a series of decision making processes. At each time step t, the navigation environment presents an image view $\boldsymbol{v}_t$. With reference to the instruction $\mathcal{X}$ and the visual view $\boldsymbol{v}_t$, the navigator is expected to choose an action $a_t \in \mathcal{A}$. The action set $\mathcal{A}$ for urban environment navigation usually contains four actions, namely *turn left*, *turn right*, *go forward*, and *stop*.

3.2 Overview

Our Multimodal Text Style Transfer (MTST) learning mainly consists of two modules, namely the *multimodal text style transfer model* and the *VLN Transformer*. Fig. 2 provides an overview of our MTST approach. We use the multimodal text style transfer model to narrow the gap between the human-annotated instructions for the outdoor navigation task and the machine-generated instructions in the external resources. The multimodal text style transfer model is trained on the dataset for outdoor navigation, and it learns to infer style-modified instructions for trajectories in the external resources. The VLN Transformer is the navigation agent that generates actions for the outdoor VLN task. It is trained with a two-stage training pipeline. We first pre-train the VLN Transformer on the external resources with the style-modified instructions and then fine-tune it on the outdoor navigation dataset.

Source	Instruction
Google Maps API	Head northwest on E 23rd St toward 2nd Ave. Turn left at the 2nd cross street onto 3rd Ave.
Human Annotator	Orient yourself so you are facing the same as the traffic on the 4 lane road. Travel down this road until the first intersection. Turn left and go down this street with the flow of traffic. You'll see a black and white stripped awning on your right as you travel down the street.

Table 1: For the outdoor VLN task, the instructions provided by Google Maps API is distinct from the instructions written by human annotators.

3.3 Multimodal Text Style Transfer Model

Instruction Style The navigation instructions vary across different outdoor VLN datasets. As shown in Table 1, the instructions generated by Google Maps API is template-based and mainly consists of street names and directions. In contrast, human-annotated instructions for the outdoor VLN task emphasize the visual environment's attributes as navigation targets. It frequently refers to objects in the panorama, such as traffic lights, cars, awnings, etc. The goal of conducting multimodal text style transfer is to inject more object-related information in the surrounding navigation environment to the machine-generated instruction while keeping the correct guiding signals.

Masking-and-Recovering Scheme The multimodal text style transfer model is trained with a "masking-and-recovering" (Zhu et al., 2019; Liu et al., 2019; Donahue et al., 2020; Huang et al., 2020a) scheme to inject objects that appeared in the panorama into the instructions. We mask out certain portions in the instructions and try to recover the missing contents with the help of the remaining instruction skeleton and the paired trajectory. To be specific, we use NLTK (Bird et al., 2009) to mask out the object-related tokens in the human-annotated instructions, and the street names

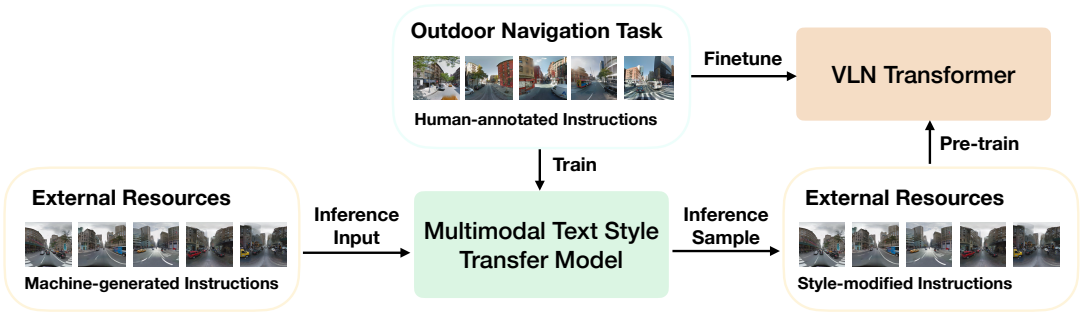


Figure 2: An overview of the Multimodal Text Style Transfer (MTST) learning approach for vision-and-language navigation in real-life urban environments. Details are described in Section 3.2.

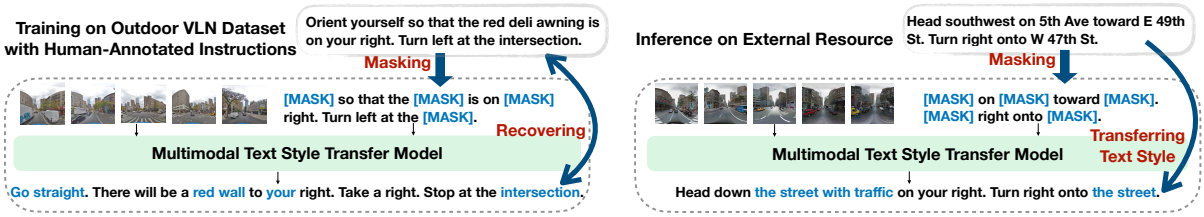


Figure 3: An example of the training and inference process of the multimodal text style transfer model. During training, we mask out the objects in the human-annotated instructions to get the instruction template. The model takes both the trajectory and the instruction skeleton as input, and the training objective is to recover the instructions with objects. When inferring new instructions for external trajectories, we mask the street names in the original instructions and prompt the model to generate new object-grounded instructions.

in the machine-generated instructions[2]. Multiple tokens that are masked out in a row will be replaced by a single [MASK] token. We aim to maintain the correct guiding signals for navigation after the style transfer process. Tokens that provide guiding signals, such as “turn left" or “take a right", will not be masked out. Fig. 3 provides an example of the “masking-and-recovering" process during training and inferring.

Model Structure Fig. 3 illustrates the input and expected output of our multimodal text style transfer model. We build the multimodal text style transfer model upon the Speaker model proposed by Fried et al. (2018). On top of the visual-attention-based LSTM (Hochreiter and Schmidhuber, 1997) structure in the Speaker model, we inject the textual attention of the masked instruction skeleton $\mathcal{X}'$ to the encoder, which allows the model to attend to original guiding signals.

The encoder takes both the visual and textual inputs, which encode the trajectory and the masked instruction skeletons. To be specific, each visual view in the trajectory is represented as a feature vector $\boldsymbol{v}' = [\boldsymbol{v}'_v; \boldsymbol{v}'_\alpha]$, which is the concatenation of the visual encoding $\boldsymbol{v}'_v \in \mathbb{R}^{512}$ and the orientation encoding $\boldsymbol{v}'_\alpha \in \mathbb{R}^{64}$. The visual encoding $\boldsymbol{v}'_v$ is the output of the last but one layer of the RESNET18 (He et al., 2016) of the current view. The orientation encoding $\boldsymbol{v}'_\alpha$ encodes current heading α by repeating vector $[sin\alpha, cos\alpha]$ for 32 times, which follows Fried et al. (2018). As described in section 3.4, the feature matrix of a panorama is the concatenation of eight projected visual views.

In the multimodal style transfer encoder, we use a soft-attention module (Vaswani et al., 2017) to calculate the grounded visual feature $\hat{\boldsymbol{v}}_t$ for current view at step t:

$$attn_{v_{t,i}} = softmax((\boldsymbol{W}_v \boldsymbol{h}_{t-1})^T \boldsymbol{v}'_i) \quad (1)$$

$$\hat{\boldsymbol{v}}_t = \sum_{i=1}^{8} = attn_{v_{t,i}} \boldsymbol{v}'_i \quad (2)$$

where $\boldsymbol{h}_{t-1}$ is the hidden context of previous step, $\boldsymbol{W}_v$ refers to the learnable parameters, and $attn_{v_{t,i}}$ is the attention weight over the i_{th} slice of view $\boldsymbol{v}'_i$ in current panorama.

We use full-stop punctuations to split the input text into multiple sentences. The rationale is to enable alignment between the street views and the semantic guidance in sub-instructions. For each sentence in the input text, the textual encoding $\boldsymbol{s}'$

[2]We masked out the tokens with the following part-of-speech tags: [JJ, JJR, JJS, NN, NNS, NNP, NNPS, PDT, POS, RB, RBR, RBS, PRP$, PRP, MD, CD]

is the average of all the tokens' word embedding in the current sentence. We also use a soft-attention modules to calculate the grounded textual feature $\hat{\boldsymbol{s}}_t$ at current step t:

$$attn_{s_{t,j}} = softmax((\boldsymbol{W}_s \boldsymbol{h}_{t-1})^T \boldsymbol{s}'_j) \quad (3)$$

$$\hat{\boldsymbol{s}}_t = \sum_{j=1}^{M} attn_{s_{t,j}} \boldsymbol{s}'_j \quad (4)$$

where $\boldsymbol{W}_s$ refers to the learnable parameters, $attn_{s_{t,j}}$ is the attention weight over the j_{th} sentence encoding $\boldsymbol{s}'_j$ at step t, and M denotes the maximum sentence number in the input text. The input text for the multimodal style transfer encoder is the instruction template $\mathcal{X}'$.

Based on the grounded visual feature $\hat{\boldsymbol{v}}_t$, the grounded textual feature $\hat{\boldsymbol{s}}_t$ and the visual view feature $\boldsymbol{v}'_t$ at current timestamp t, the hidden context can be given as:

$$\boldsymbol{h}_t = LSTM([\hat{\boldsymbol{v}}_t; \hat{\boldsymbol{s}}_t; \boldsymbol{v}'_t]) \quad (5)$$

Training Objectives We train the multimodal text style transfer model in the teacher-forcing manner (Williams and Zipser, 1989). The decoder generates tokens auto-regressively, conditioning on the masked instruction template $\mathcal{X}'$, and the trajectory. The training objective is to minimize the following cross-entropy loss:

$$\mathcal{L}(x_1, x_2, \ldots, x_n | \mathcal{X}', \boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_N) = -\log \prod_{j=1}^{n} P(x_j | x_1, ..., x_{j-1}, \mathcal{X}', \boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_N) \quad (6)$$

where $x_1, x_2, \ldots, x_n$ denotes the tokens in the original instruction $\mathcal{X}$, n is the total token number in $\mathcal{X}$, and N denotes the maximum view number in the trajectory.

3.4 VLN Transformer

The VLN Transformer is the navigation agent that generates actions in the outdoor VLN task. As illustrated in Fig. 4, our VLN Transformer is composed of an instruction encoder, a trajectory encoder, a cross-modal encoder that fuses the modality of the instruction encodings and trajectory encodings, and an action predictor.

Instruction Encoder The instruction encoder is a pre-trained uncased BERT-base model (Devlin et al., 2019). Each piece of navigation instruction is split into multiple sentences by the full-stop punctuations. For the i_{th} sentence $s_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,l_i}\}$ that contains l_i tokens, its sentence embedding $\boldsymbol{h}^s_i$ is calculated as:

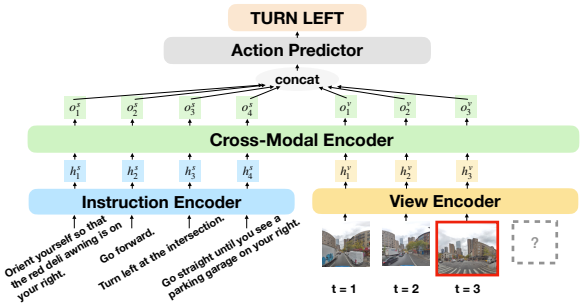


Figure 4: Overview of the VLN Transformer. In this example, the VLN Transformer predicts to take a left turn for the visual scene at $t = 3$.

$$\boldsymbol{w}_{i,j} = \mathcal{BERT}(x_{i,j}) \in \mathbb{R}^{768} \quad (7)$$

$$\boldsymbol{h}^s_i = \mathcal{FC}(\frac{\sum_{j=1}^{l_i} \boldsymbol{w}_{i,j}}{l_i}) \in \mathbb{R}^{256} \quad (8)$$

where $\boldsymbol{w}_{i,j}$ is the word embedding for $x_{i,j}$ generated by BERT, and $\mathcal{FC}$ is a fully-connected layer.

View Encoder We use the view encoder to retrieve embeddings for the visual views at each time step. Following Chen et al. (2019), we embed each panorama $\mathbf{I}_t$ by slicing it into eight images and projecting each image from an equirectangular projection to a perspective projection. Each of the projected image of size 800×460 will be passed through the RESNET18 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015). We use the output of size $128 \times 100 \times 58$ from the fourth to last layer before classification as the feature for each slice. The feature map for each panorama is the concatenation of the eight image slices, which is a single tensor of size $128 \times 100 \times 464$. We center the feature map according to the agent's heading α_t at timestamp t. We crop a $128 \times 100 \times 100$ sized feature map from the center and calculate the mean value along the channel dimension. The resulting 100×100 features is regarded as the current panorama feature $\hat{\mathbf{I}}_t$ for each state. Following Mirowski et al. (2018), we then apply a three-layer convolutional neural network on $\hat{\mathbf{I}}_t$ to extract the view features $\boldsymbol{h}^v_t \in \mathbb{R}^{256}$ at timestamp t.

Cross-Modal Encoder In order to navigate through complicated real-world environments, the agent needs to grasp a proper understanding of the natural language instructions and the visual views jointly to choose proper actions for each state. Since the instructions and the trajectory

lies in different modalities and are encoded separately, we introduce the cross-modal encoder to fuse the features from different modalities and jointly encode the instructions and the trajectory. The cross-modal encoder is an 8-layer Transformer encoder (Vaswani et al., 2017) with mask. We use eight self-attention heads and a hidden size of 256.

In the teacher-forcing training process, we add a mask when calculating the multi-head self-attention across different modalities. By masking out all the future views in the ground-truth trajectory, the current view $\boldsymbol{v}_t$ is only allowed to refer to the full instructions and all the previous views that the agent has passed by, which is $[\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, \ldots, \boldsymbol{h}_M^s; \boldsymbol{h}_1^v, \boldsymbol{h}_2^v, \ldots, \boldsymbol{h}_{t-1}^v]$, where M denotes the maximum sentence number.

Since the Transformer architecture is based solely on attention mechanism and thus contains no recurrence or convolution, we need to inject additional information about the relative or absolute position of the features in the input sequence. We add a learned segment embedding to every input feature vector specifying whether it belongs to the sentence encodings or the view encodings. We also add a learned position embedding to indicate the relative position of the sentences in the instruction sequence or the trajectory sequence's views.

Action Predictor The action predictor is a fully-connected layer. It takes the concatenation of the cross-modal encoder's output up to the current timestamp t as input, and predicts the action a_t for view $\boldsymbol{v}_t$:

$$\boldsymbol{h}_{concat} = \boldsymbol{h}_1^s || \ldots || \boldsymbol{h}_M^s || \boldsymbol{h}_1^v || \ldots || \boldsymbol{h}_t^v \quad (9)$$

$$a_t = argmax(\mathcal{FC}(\mathcal{T}(\boldsymbol{h}_{concat}))) \quad (10)$$

where $\mathcal{FC}$ is a fully-connected layer in the action predictor, and $\mathcal{T}$ refers to the Transformer operation in the cross-modal encoder. During training, we use the cross-entropy loss for optimization.

4 Experiments

4.1 Datasets

Outdoor VLN Dataset For the outdoor VLN task, we conduct experiments on the Touchdown dataset (Chen et al., 2019; Mehta et al., 2020), which is designed for navigation in realistic urban environments. Based on Google Street View[3], Touchdown's navigation environment encompasses 29,641 Street View panoramas of the Manhattan area in New York City, which are connected by 61,319 undirected edges. The dataset contains 9,326 trajectories for the navigation task, and each trajectory is paired with a human-written instruction. The training set consists of 6,526 samples, while the development set and the test set are made up of 1,391 and 1,409 samples, respectively.

External Resource We use the StreetLearn dataset as the external resource for the outdoor VLN task (Mirowski et al., 2018). The StreetLearn dataset is another dataset for navigation in real-life urban environments based on Google Street View. StreetLearn contains 114k panoramas from New York City and Pittsburgh. In the StreetLearn navigation environment, the graph for New York City contains 56k nodes and 115k edges, while the graph for Pittsburgh contains 57k nodes and 118k edges. The StreetLearn dataset contains 580k samples in the Manhattan area and 8k samples in the Pittsburgh area for navigation.

While the StreetLearn dataset's trajectory contains more panorama along the way on average, the paired instructions are shorter than the Touchdown dataset. We extract a sub-dataset *Manh-50* from the original large scale StreetLearn dataset for the convenience of conducting experiments. Manh-50 consists of navigation samples in the Manhattan area that contains no more than 50 panoramas in the whole trajectory, containing 31k training samples. We generate style-transferred instructions for the Manh-50 dataset, which serves as an auxiliary dataset, and will be used to pre-train the navigation models. More details can be found in the appendix.

4.2 Evaluation Metrics

We use the following metrics to evaluate VLN performance: (1) *Task Completion (TC)*: the accuracy of completing the navigation task correctly. Following Chen et al. (2019), the navigation result is considered correct if the agent reaches the specific goal or one of the adjacent nodes in the environment graph. (2) *Shortest-Path Distance (SPD)*: the mean distance between the agent's final position and the goal position in the environment graph. (3) *Success weighted by Edit Distance (SED)*: the normalized Levenshtein edit distance between the path predicted by the agent and the reference path, which is constrained only to the successful navigation. (4) *Coverage weighted by Length Score (CLS)*: a measurement of the fidelity of the agent's path

[3] https://developers.google.com/maps/documentation/streetview/intro

with regard to the reference path. (5) *Normalized Dynamic Time Warping (nDTW)*: the minimized cumulative distance between the predicted path and the reference path, normalized by the reciprocal of the square root of the reference path length. The value is rescaled by taking the negative exponential of the normalized value. (6) *Success weighted Dynamic Time Warping (SDTW)*: the nDTW value where the summation is only over the successful navigation.

TC, SPD, and SED are defined by Chen et al. (2019). CLS is defined by Jain et al. (2019). nDTW and SDTW are originally defined by Ilharco et al. (2019), in which nDTW is normalized by the length of the reference path. We adjust the normalizing factor to be the reciprocal of the square root of the reference path length for length invariance (Mueen and Keogh, 2016). In case the reference trajectories length has a salient variance, our modification to the normalizing factor made the nDTW and SDTW scores invariant to the reference length.

4.3 Results and Analysis

In this section, we report the outdoor VLN performance and the quality of the generated instructions to validate the effectiveness of our MTST learning approach. We compare our VLN Transformer with the baseline model and discuss the influence of pre-training on external resources with/without instruction style transfer.

Outdoor VLN Performance We compare our VLN Transformer with RCONCAT (Chen et al., 2019; Mirowski et al., 2018) and GA (Chen et al., 2019; Chaplot et al., 2018) as baseline models. Both baseline models encode the trajectory and the instruction in an LSTM-based manner and use supervised training with Hogwild! (Recht et al., 2011). Table 2 presents the navigation results on the Touchdown validation and test sets, where VLN Transformer performs better than RCONCAT and GA on most metrics with the exception of SPD and CLS.

Pre-training the navigation models on Manh-50 with template-based instructions can partially improve navigation performance. For all three agent models, the scores related to successful cases—such as TC, SED, and SDTW—witness a boost after being pre-trained on vanilla Manh-50. However, the instruction style difference between Manh-50 and Touchdown might misguide the agent in the pre-training stage, resulting in a performance drop on SPD for our VLN Transformer model.

In contrast, our MTST learning approach can better utilize external resources and further improve navigation performance. Pre-training on Manh-50 with style-modified instructions can stably improve the navigation performance on all the metrics for both the RCONCAT model and the VLN Transformer. This also indicates that our MTST learning approach is model-agnostic.

Table 4 compares the SPD values on success and failure navigation cases. In the success cases, VLN Transformer has better SPD scores, which is aligned with the best SED results in Table 2. Our model's inferior SPD results are caused by taking longer paths in failure cases, which also harms the fidelity of the generated path and lowers the CLS scores. Nevertheless, every coin has two sides, and exploring more areas when getting lost might not be a complete bad behavior for the navigation agent. We leave this to future study.

Multimodal Text Style Transfer in VLN We attempt to reveal each component's effect in the multimodal text style transfer model. We pre-train the VLN Transformer with external trajectories and instructions generated by different models, then fine-tune it on the TouchDown dataset.

According to the navigation results in Table 3, the instructions generated by the Speaker model misguide the navigation agent, indicating that relying solely on the Speaker model cannot reduce the gap between different instruction styles. Adding textual attention to the Speaker model can slightly improve the navigation results, but still hinders the agent from navigating correctly. The style-modified instructions improve the agent's performance on all the navigation metrics, suggesting that our Multimodal Text Style Transfer learning approach can assist the outdoor VLN task.

Quality of the Generated Instruction We evaluate the quality of instructions generated by the Speaker and the MTST model. We utilize five automatic metrics for natural language generation to evaluate the quality of the generated instructions, including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Elliott and Keller, 2013), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). In addition, we calculate the guiding signal match rate (MR) by comparing the appearance of "turn left" and "turn right". If the generated instruction contains the

Model	Dev Set						Test Set					
	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	SDTW ↑	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	SDTW ↑
RCONCAT	10.6	20.4	10.3	48.1	22.5	9.8	11.8	20.4	11.5	47.9	22.9	11.1
+*M-50*	11.8	19.1	11.4	48.7	23.1	10.9	12.1	19.4	11.8	49.4	24.0	11.3
+*M-50* +style	11.9	19.9	11.5	48.9	23.8	11.1	12.6	20.4	12.3	48.0	23.9	11.8
GA	12.0	18.7	11.6	51.9	25.2	11.1	11.9	19.0	11.5	51.6	24.9	10.9
+*M-50*	12.3	**18.5**	11.8	**53.7**	26.2	11.3	13.1	**18.4**	12.8	**54.2**	26.8	12.1
+*M-50* +style	12.9	**18.5**	12.5	52.8	26.3	11.9	13.9	**18.4**	13.5	53.5	27.5	12.9
VLN Transformer	14.0	21.5	13.6	44.0	23.0	12.9	14.9	21.2	14.6	45.4	25.3	14.0
+*M-50*	14.6	22.3	14.1	45.6	25.0	13.4	15.5	21.9	15.4	45.9	26.1	14.2
+*M-50* +style	**15.0**	20.3	**14.7**	50.1	**27.0**	**14.2**	**16.2**	20.8	**15.7**	50.5	**27.8**	**15.0**

Table 2: Navigation results on the outdoor VLN task. *+M-50* denotes pre-training with vanilla Manh-50 which contains machine-generated instructions; in the *+style* setting, the model is pre-trained with Manh-50 trajectories and style-modified instructions that are generated by our MTST model.

Model	Dev Set						Test Set					
	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	SDTW ↑	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	SDTW ↑
VLN Transformer +*M-50*	14.6	22.3	14.1	45.6	25.0	13.4	15.5	21.9	15.4	45.9	26.1	14.2
+speaker	7.6	26.2	7.3	34.6	14.6	7.0	8.3	25.4	8.0	36.3	15.9	7.7
+text_attn	11.7	**20.1**	11.3	46.3	23.2	10.7	11.8	**20.5**	11.5	47.3	23.2	11.0
+style	**15.0**	20.3	**14.7**	**50.1**	**27.0**	**14.2**	**16.2**	20.8	**15.7**	**50.5**	**27.8**	**15.0**

Table 3: Ablation study of the multimodal text style transfer model on the outdoor VLN task. In the *+speaker* setting, the instructions used in pre-training are generated by the Speaker (Fried et al., 2018), which only attends to the visual input; *+text_attn* denotes that we add a textual attention module to the Speaker to attend to both the visual input and the machine-generated instructions provided by Google Maps API.

Model	Dev Set		Test Set	
	S_SPD↓	F_SPD↓	S_SPD↓	F_SPD↓
RCONCAT	0.64	22.68	0.67	23.06
+*M-50*	0.68	21.53	0.69	21.97
+*M-50* +style	0.66	22.48	0.69	23.21
GA	0.65	21.15	0.66	21.41
+*M-50*	0.70	**20.95**	0.77	**21.09**
+*M-50* +style	0.65	21.11	0.70	21.26
VLN Transformer	0.66	24.92	0.63	24.84
+*M-50*	0.67	25.94	0.63	25.77
+*M-50* +style	**0.59**	23.72	**0.62**	24.67

Table 4: *S_SPD* and *F_SPD* denotes the average SPD value on success and failure cases respectively.

same number of guiding signals in the same order as the ground truth instruction, then this instruction pair is considered to be matched. We also calculate the number of different infilled tokens (#infill) in the generated instruction[4]. This reflects the model's ability to inject object-related information during style transferring. Among the 9,326 trajectories in the Touchdown dataset, 9,000 are used to train the MTST model, while the rest form the validation set.

[4]We regard tokens with the following part-of-speech tags as infilled tokens: [JJ, JJR, JJS, NN, NNS, NNP, NNPS, PDT, POS, RB, RBR, RBS, PRP$, PRP, MD, CD]

Model	BLEU	METEOR	ROUGE_L	CIDEr	SPICE	MR	#infill
Speaker	15.1	20.6	22.2	1.4	20.7	8.3	160
Text_Attn	23.8	23.3	29.6	10.0	24.6	35.7	182
MTST	**30.6**	**28.8**	**39.7**	**27.8**	**30.6**	**46.7**	**308**

Table 5: Quantitative evaluation of the instructions generated by Speaker, Speaker with textual attention and our MTST model.

We report the quantitative results on the validation set in Table 5. After adding textual attention to the Speaker, the evaluation performance on all seven metrics improved. Our MTST model scores the highest on all seven metrics, which indicates that the "masking-and-recovering" scheme is beneficial for the multimodal text style transfer process. The results validate that the MTST model can generate higher quality instructions, which refers to more visual objects and provide more matched guiding signals.

Human Evaluation We invite human judges on Amazon Mechanical Turk to evaluate the quality of the instructions generated by different models. We conduct a pairwise comparison, which covers 170 pairs of instructions generated by Speaker, Speaker with textual attention, and our MTST model. The instruction pairs are sampled from the Touchdown

Choice (%)	MTST *vs* Speaker			MTST *vs* Text_Attn			Speaker *vs* Text_Attn		
	MTST	Speaker	Tie	MTST	Text_Attn	Tie	Speaker	Text_Attn	Tie
Better describes the street view	**67.9**	22.8	9.3	**44.3**	35.8	19.9	28.2	**62.7**	9.1
More aligned with the ground truth	**64.6**	26.8	8.6	**37.6**	33.9	28.5	25.3	**62.5**	12.2

Table 6: Human evaluation results of the instructions generated by Speaker, Speaker with textual attention and our MTST model with pairwise comparisons.

validation set. Each pair of instructions, together with the ground truth instruction and the gif that illustrates the navigation street view, is presented to 5 annotators. The annotators are asked to make decisions from the aspect of guiding signal correctness and instruction content alignment. Results in Table 6 show that annotators think the instructions generated by our MTST model better describe the street view and is more aligned with the ground-truth instructions.

Case Study We demonstrate case study results to illustrate the performance of our Multimodal Text Style Transfer learning approach. Fig. 5 provides two showcases of the instruction generation results. As listed in the charts, the instructions generated by the vanilla Speaker model have a poor performance in keeping the guiding signals in the ground truth instructions and suffer from hallucinations, which refers to objects that have not appeared in the trajectory. The Speaker with textual attention can provide guidance direction. However, the instructions generated in this manner does not utilize the rich visual information in the trajectory. On the other hand, the instructions generated by our multimodal text style transfer model inject more object-related information ("the light", "scaffolding") in the surrounding navigation environment to the StreetLearn instruction while keeping the correct guiding signals.

Figure 5: Two showcases of the instruction generation results. The red tokens indicate incorrectly generated instructions, while the blue tokens suggest alignments with the ground truth. The orange bounding boxes show that the objects in the surrounding environment have been successfully injected into the style-modified instruction.

5 Conclusion

In this paper, we proposed the Multimodal Text Style Transfer learning approach for outdoor VLN. This learning framework allows us to utilize out-of-domain navigation samples in outdoor environments and enrich the original navigation reasoning training process. Experimental results show that our MTST approach is model-agnostic, and our MTST learning approach outperforms the baseline models on the outdoor VLN task. We believe our study provides a possible solution to mitigate the data scarcity issue in the outdoor VLN task. In future studies, we would love to explore the possibility of constructing an end-to-end framework. We will also further improve the quality of style-modified instructions, and quantitatively evaluate the alignment between the trajectory and the style-transferred instructions.

Acknowledgments

We would like to show our gratitude towards Jiannan Xiang, who kindly shares his experimental code on Touchdown, and Qi Wu, who provides valuable feedback to our initial draft. We also thank the anonymous reviewers for their thought-provoking comments. The UCSB authors were sponsored by an unrestricted gift from Google. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the sponsor.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3674–3683. IEEE Computer Society.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2018. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2819–2826. AAAI Press.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12538–12547. Computer Vision Foundation / IEEE.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: universal image-text representation learning. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2492–2501. Association for Computational Linguistics.

Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1292–1302. ACL.

Norman E. Fenton, Martin Neil, and Anthony C. Constantinou. 2020. *The Book of Why: The New Science of Cause and Effect, Judea Pearl, Dana Mackenzie. Basic Books (2018)*, volume 284.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3318–3329.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 663–670. AAAI Press.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020. Towards learning a generic agent for vision-and-language navigation via pre-training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13134–13143. IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. volume 9, pages 1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. 2020a. INSET: sentence infilling with inter-sentential transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2502–2515. Association for Computational Linguistics.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020b. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *CoRR*, abs/2004.00849.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*.

Vihan Jain, Gabriel Magalhães, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1862–1872. Association for Computational Linguistics.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 424–434. Association for Computational Linguistics.

Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha S. Srinivasa. 2019. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6741–6749. Computer Vision Foundation / IEEE.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11336–11344. AAAI Press.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. volume abs/1908.03557.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Dayiheng Liu, Jie Fu, Pengfei Liu, and Jiancheng Lv. 2019. TIGS: an inference algorithm for text infilling with gradient search. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4146–4156. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. volume abs/2002.06353.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019a. Self-monitoring navigation agent via auxiliary progress estimation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019b. The regretful agent: Heuristic-aided navigation through progress estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6732–6740. Computer Vision Foundation / IEEE.

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 259–274. Springer.

Harsh Mehta, Yoav Artzi, Jason Baldridge, Eugene Ie, and Piotr Mirowski. 2020. Retouchdown: Adding touchdown to streetlearn as a shareable resource for language grounding tasks in street view. volume abs/2001.03671.

Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. 2018. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2424–2435.

Abdullah Mueen and Eamonn J. Keogh. 2016. Extracting optimal performance from dynamic time warping. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 2129–2130. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Benjamin Recht, Christopher Ré, Stephen J. Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 693–701.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. Imagenet large scale visual recognition challenge. volume 115, pages 211–252.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7463–7472. IEEE.

Hao Tan and Mohit Bansal. 2019. LXMERT: learning cross-modality encoder representations from transformers. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5099–5110. Association for Computational Linguistics.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2610–2621. Association for Computational Linguistics.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6558–6569. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Xin Wang, Qiuyuan Huang, Asli Çelikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 6629–6638. Computer Vision Foundation / IEEE.

Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, volume 11220 of *Lecture Notes in Computer Science*, pages 38–55. Springer.

Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020. Multi-modality cross attention network for image and sentence matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10938–10947. IEEE.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. volume 1, pages 270–280.

Jiannan Xiang, Xin Wang, and William Yang Wang. 2020. Learning to stop: A simple yet effective approach to urban vision-language navigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 699–707. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7298–7309.

Yubo Zhang, Hao Tan, and Mohit Bansal. 2020. Diagnosing the environment bias in vision-and-language navigation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 890–897. ijcai.org.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.

Chen Zheng, Quan Guo, and Parisa Kordjamshidi. 2020. Cross-modality relevance for reasoning on language and vision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7642–7651. Association for Computational Linguistics.

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *CoRR*, abs/1901.00158.

A Appendix

A.1 Dataset Comparison

Dataset	#path	#pano	#pano/path	instr_len	#sent/path	#turn/path
Touchdown	6k	26k	35.2	80.5	6.3	2.8
Manh-50	31k	43k	37.2	22.1	2.8	4.1
StreetLearn	580k	114k	29.0	28.6	4.0	13.2

Table 7: Dataset statistics. *path*: navigation path; *pano*: panorama; *instr_len*: average instruction length; *sent*: sentence; *turn*: intersection on the path.

Table 7 lists out the statistical information of the datasets used in pre-training and fine-tuning. Even though the Touchdown dataset and the StreetLearn dataset are built upon Google Street View, and both of them contain urban environments in New York City, pre-training the model with the VLN task on the StreetLearn dataset does not raise a threat of test data leaking. This is due to several causes:

First, the instructions in the two datasets are distinct in styles. The instructions in the StreetLearn dataset is generated by Google Maps API, which is template-based and focuses on street names. However, the instructions in the Touchdown dataset are created by human annotators and emphasize the visual environment's attributes as navigational cues. Moreover, as reported by Mehta et al. (2020), the panoramas in the two datasets have little overlaps. In addition, Touchdown instructions constantly refer to transient objects such as cars and bikes, which might not appear in a panorama from a different time. The different granularity of the panorama spacing also leads to distinct panorama distributions of the two datasets.

A.2 Training Details

We use Adam optimizer (Kingma and Ba, 2015) to optimize all the parameters. During pre-training on the StreetLearn dataset, the learning rate for the RCONCAT model, GA model, and the VLN Transformer is 2.5×10^{-4}. We fine-tune BERT separately with a learning rate of 1×10^{-5}. We pre-train RCONCAT and GA for 15 epochs and pre-train the VLN Transformer for 25 epochs.

When training or fine-tuning on the Touchdown dataset, the learning rate for RCONCAT and GA is 2.5×10^{-4}. For the VLN Transformer, the learning rate to fine-tune BERT is initially set to 1×10^{-5}, while the learning rate for other parameters in the model is initialized to be 2.5×10^{-4}. The learning rate for VLN Transformer will decay. The batch size for RCONCAT and GA is 64, while the VLN Transformer uses a batch size of 30 during training.

Model	TC ↑	SPD ↓	SED ↑	CLS ↑	nDTW ↑	SDTW ↑
no split	9.6	21.8	9.3	46.1	20.0	8.7
split	**13.6**	**20.5**	**13.1**	**47.6**	**24.0**	**12.6**

Table 8: Ablation results of the VLN Transformer's instruction split on Touchdown dev set. In *split* setting, the instruction is split into multiple sentences before being encoded by the instruction encoder, while *no split* setting encodes the whole instruction without splitting.

A.3 Split Instructions vs. No Split

We compare VLN Transformer performance with and without splitting the instructions into sentences during encoding. Results in Table 8 show that breaking the instructions into multiple sentences allows the visual views and the guiding signals in sub-instructions to attend to each other during cross-modal encoding fully. Such cross-modal alignments lead to betters navigation performance.

A.4 Amazon Mechanical Turk

We use AMT for human evaluation when evaluating the quality of the instructions generated by different models. The survey form for head-to-head comparisons is shown in Figure 6.

Watch the gif that shows the change of the streetviews, then read the ground truth instruction that describes the navigation process.

Ground Truth: Go in the direction of traffic, with a bridge ahead of you. Go straight through intersection. (You may have to go right to go straight.) At the next intersection, see the bridge tunnel on the right. Go straight through the intersection. At the end of the road, with water in front of you, go in the intersection and turn to come back down the road you are on. When you get back to the bridge tunnel intersection, go right just far enough across the intersection to turn and go left in the far lane of traffic. There is a brick building to your right, and chain and post fences to your left. Go past the first yellow sign with black people, and stop next to the second yellow sign with black people.

A high-quality instruction should be **informative (address to the objects in the surrounding)**, and should be able to **provide correct guiding signals (turn left/right)**.

Please read the two instructions below, and select which instruction is more aligned to the ground truth instruction, and describes the streetview in the gif better.

A. Go in the direction of traffic, with the flow of traffic in front of you. Go right at the intersection. You'll have a lot of blue bikes on the right. Go to the next intersection and turn right. When you turn, there will be a green awning on the right and white and yellow umbrellas on the right. Go straight through the first intersection you come to, and at the second intersection, turn left. Go to the next intersection and turn left. Stop just past the crosswalk, and turn to face the building with the green awning, you'll see a building with red trim, and red doors.

B. Go in the direction that puts the green construction wall on your right. Go straight through the first intersection you come to. At the next intersection, turn right. Go to the next intersection and turn right again. Go straight through the next intersection. When you get to the next intersection, go straight through it. Stop just before the crosswalk of the next intersection.

Which instruction better describe the streetview? ○ A ○ B ○ Tie

Which instruction is more aligned with the ground truth? ○ A ○ B ○ Tie

Submit

Figure 6: Pairwise comparison form for human evaluation on AMT.