Airbert: In-domain Pretraining for Vision-and-Language Navigation
Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, Cordelia Schmid

To cite this version:

HAL Id: hal-03470013
https://hal.science/hal-03470013
Submitted on 8 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Airbert: In-domain Pretraining for Vision-and-Language Navigation

Pierre-Louis Guhur¹, Makarand Tapaswi², Shizhe Chen¹, Ivan Laptev¹, Cordelia Schmid¹
¹Inria, École normale supérieure, CNRS, PSL Research University, Paris, France
²IIIT Hyderabad, India

Email: pierre-louis.guhur@inria.fr  Website: https://airbert-vln.github.io

Abstract

Vision-and-language navigation (VLN) aims to enable embodied agents to navigate in realistic environments using natural language instructions. Given the scarcity of domain-specific training data and the high diversity of image and language inputs, the generalization of VLN agents to unseen environments remains challenging. Recent methods explore pretraining to improve generalization, however, the use of generic image-caption datasets or existing small-scale VLN environments is suboptimal and results in limited improvements. In this work, we introduce BnB, a large-scale and diverse in-domain VLN dataset. We first collect image-caption (IC) pairs from hundreds of thousands of listings from online rental marketplaces. Using IC pairs we next propose automatic strategies to generate millions of VLN path-instruction (PI) pairs. We further propose a shuffling loss that improves the learning of temporal order inside PI pairs. We use BnB to pretrain our Airbert model that can be adapted to discriminative and generative settings and show that it outperforms state of the art for Room-to-Room (R2R) navigation and Remote Referring Expression (REVERIE) benchmarks. Moreover, our in-domain pretraining significantly increases performance on a challenging few-shot VLN evaluation, where we train the model only on VLN instructions from a few houses.

1. Introduction

In vision-and-language navigation (VLN), an agent is asked to navigate in home environments following natural language instructions [3, 5]. This task is attractive to many real-world applications such as domestic robotics and personal assistants. However, given the high diversity of VLN data across environments and the difficulty of the manual collection and annotation of VLN training data at scale, the performance of current methods remains limited, especially for previously unseen environments [49].

Our work is motivated by significant improvements in vision and language pretraining [2, 9, 23, 24, 25, 38], where deep transformer models [42] are trained via self-supervised proxy tasks [10] using large-scale, automatically harvested image-text datasets [32, 36]. Such pretraining enables learning transferable multi-modal representations achieving state-of-the-art performance in various vision and language tasks. Similarly, with the goal of learning an embodied agent that generalizes, recent works [13, 16, 22, 28] have explored different pretraining approaches for VLN tasks.

In [13, 16], annotated path-instruction pairs are augmented with a speaker model that generates instructions for random unseen paths. However, as these paths originate from a small set of 61 houses used during training, they are limited in visual diversity. The limited pretraining
environments do not equip agents with visual understanding abilities that enable generalization to unseen houses, see Fig. 1. To address this problem, VLN-BERT [28] proposes to pretrain the agent on generic image-caption datasets that are abundant and cover diverse visio-linguistic knowledge. However, these image-caption pairs are quite different from the dynamic visual stream (path) and navigable instructions observed by a VLN agent. Such out-of-domain pretraining, although promising, only brings limited gains to the navigation performance. Besides the above limitations, existing pretraining methods do not place much emphasis on temporal reasoning abilities in their proxy tasks such as one-step action prediction [13] and path-instruction pairing [28], while such reasoning is important to a sequential decision making task like VLN. As a result, even if performance in downstream tasks is improved, the pretrained models may still be brittle. For example, a simple corruption of instructions by swapping noun phrases within the instruction, or replacing them with other nouns, leads to significant confusion as models are unable to pick the correct original pair.

In this paper, we explore a different data source and proxy tasks to address the above limitations in pretraining a generic VLN agent. Though navigation instructions are rarely found on the Internet, image-caption pairs from home environments are abundant in online marketplaces (e.g. Airbnb), which include images and descriptions of rental listings. We collect BnB, a new large-scale dataset with 1.4M indoor images and 0.7M captions. First, we show that in-domain image-caption pairs bring additional benefits for downstream VLN tasks when applied with generic web data [28]. In order to further reduce the domain gap between the BnB pretraining and the VLN task, we present an approach to transform static image-caption pairs into visual paths and navigation-like instructions (Fig. 1 bottom), leading to large additional performance gains. We also propose a shuffling loss that improves the model’s temporal reasoning abilities by learning a temporal alignment between a path and the corresponding instruction.

Our pretrained model, Airbert, is a generic transformer backbone that can be readily integrated in both discriminative VLN tasks such as path-instruction compatibility prediction [28] and generative VLN tasks [15] in R2R navigation [5] and REVERIE remote referring expression [34]. We achieve state-of-the-art performance on these VLN tasks with our pretrained model. Beyond the standard evaluation, our in-domain pretraining opens an exciting new direction of one/few-shot VLN where the agent is trained on examples only from one/few environment(s) and expected to generalize to other unseen environments.

In summary, the contributions of this work are threefold. (1) We collect a new large-scale in-domain dataset, BnB, to promote pretraining for vision-and-language navigation tasks. (2) We curate the dataset in different ways to reduce the distribution shift between pretraining and VLN and also propose the shuffling loss to improve temporal reasoning abilities. (3) Our pretrained Airbert can be plugged into generative or discriminative architectures and achieves state-of-the-art performance on R2R and REVERIE datasets. Moreover, our model generalizes well under a challenging one/few-shot VLN evaluation, truly highlighting the capabilities of our learning paradigm.

2. Related work

Vision-and-language navigation. VLN [5] has received significant attention with a large number of followup tasks introduced in recent years [3, 8, 19, 21, 30, 31, 34, 37, 41]. Early days of VLN saw the use of sequence-to-sequence LSTMs to predict low-level actions [5] or high-level directions in a panoramic action space [11]. Different attention mechanisms [26, 33] are proposed to improve cross-modal alignment. Various reinforcement learning based training algorithms [40, 45, 47, 48] and searching algorithms in inference [11, 26, 27] have also been explored to improve the VLN performance.

To improve an agent’s generalization to unseen environments, data augmentation is performed by using a speaker model [11] that generates instructions for random paths in seen environments, and environment dropout [40] is used to mimic new environments. While pretraining LSTMs for transferable representations is adopted by [16], recently, there has been a shift towards transformer models [13] to learn generic multimodal representations. This is further extended to a recurrent model that significantly improves sequential action prediction [15]. However, the limited environments in pretraining [13, 16] constrain the generalization ability to unseen scenarios. Most related to this work, VLN-BERT [28] transfers knowledge from abundant, but out-of-domain image-text data to improve path-instruction matching. In this work, we not only create a large-scale, in-domain BnB dataset, but also propose effective pretraining strategies to mitigate the domain-shift between weably crawled image-text pairs and VLN data.

Large-scale visio-linguistic pretraining. Thanks to large-scale image-caption pairs automatically collected from the web [29, 32, 35, 36], visio-linguistic pretraining (VLP) has made great breakthroughs in recent years. Several VLP models [9, 23, 24, 39] have been proposed based on the transformer architecture [42]. These models are often pretrained with self-supervised objectives akin to those in BERT [10]: masked language modeling, masked region modeling and vision-text pairing. Fine-tuning them on downstream datasets achieves state-of-the-art performance on various VL tasks [6, 18, 46, 44]. While such pretraining focuses on learning correlations between vision and text, it is not designed for sequential decision making as required.
in embodied VLN. The goal of this work is not to improve VLP architectures but to present in-domain training strategies that lead to performance improvements for VLN tasks.

3. BnB Dataset

Hosts that rent places on online marketplaces often upload attractive and unique photos along with descriptions. One such marketplace, Airbnb, has 5.6M listings from over 100K cities all around the world [1]. We propose to use this abundant and curated data for large-scale in-domain VLN pretraining. In this section, we first describe how we collect image-caption pairs from Airbnb. Then, we propose methods to transform images and captions into VLN-like path-instruction pairs to reduce the domain gap between webly crawled image-caption pairs and VLN tasks (see Fig. 2).

3.1. Collecting BnB Image-Caption Pairs

Collection process. We restrict our dataset to listings from the US (about 10% of Airbnb) to ensure high quality English captions and visual similarity with Matterport environments [7]. The data collection proceeds as follows: (1) obtain a list of locations from Wikipedia; (2) find listings in these locations by querying the Airbnb search engine; (3) download listings and their metadata; (4) remove outdoors images\(^3\) as classified by a ResNet model pretrained on Places365 [50]; and (5) remove invalid image captions such as emails, URLs and duplicates.

Statistics. We downloaded almost 150k listings and their metadata (1/4 of the listings in the US) in step 3, leading to over 3M images and 1M captions. After data cleaning with steps 4 and 5, we obtain 713K image-caption pairs and 676K images without captions. Table 1 compares our BnB dataset to other datasets used in previous works for VLN (pre-)training. It is larger than R2R [5], REVERIE [34] and includes a large diversity of rooms and objects, which is not the case for Conceptual Captions [36]. We posit that such in-domain data is crucial to deal with the data scarcity challenge in VLN environments as illustrated in Fig. 1. We use 95% of our BnB dataset for training and the remaining 5% for validation.

Apart from images and captions, our collected listings contain structured data including a list of amenities, a general description, reviews, location, and rental price, which may offer additional applications in the future. More details about the dataset and examples are presented in the Appendix A.

3.2. Creating BnB Path-Instruction Pairs

BnB image-caption (IC) pairs are complementary to Conceptual Captions (ConCaps) as they capture diverse

\(^3\)While outdoor images may contain interesting features (e.g., a patio), we observe that removing them increases performance.

Table 1: Comparing BnB to other existing VLN datasets. The #images from Matterport environments [7] refers to the #panoramas. The speaker model [40] generates instructions for randomly selected trajectories, but is limited to panoramas from 60 training environments. Note that the data from Conceptual Captions (ConCaps) may feature some houses, but it is not the main category.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>#Envs</th>
<th>#Images</th>
<th>#Texts</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVERIE [34]</td>
<td>Matterport</td>
<td>86</td>
<td>10.6K</td>
<td>10.6K</td>
</tr>
<tr>
<td>Speaker [40]</td>
<td>Matterport</td>
<td>60</td>
<td>7.8K</td>
<td>0.2M</td>
</tr>
<tr>
<td>ConCaps [36]</td>
<td>Web images</td>
<td>-</td>
<td>3.3M</td>
<td>3.3M</td>
</tr>
<tr>
<td>BnB (ours)</td>
<td>Airbnb</td>
<td>140K</td>
<td>1.4M</td>
<td>0.7M</td>
</tr>
</tbody>
</table>

3.2.2 Augmenting Paths with Visual Contexts

In the above concatenated path, each location only contains one BnB image, and perhaps with a limited view angle as hosts may focus on objects or amenities they wish to highlight. Therefore, it lacks the panoramic visual context at each location that the agent receives in real navigation paths. Moreover, each location in the concatenated instruction is described by a unique sentence, while adjacent locations are often expressed together in one sentence in VLN instructions [14]. To address the above issues with concatenation, we propose two approaches to compose paths that have more visual context and can also leverage the abundant images without captions (denoted as captionless images).

1. Image merging extends the panoramic context of a location by grouping images from similar room categories (see Fig. 2). For example, if the image depicts a kitchen sink, it is natural to expect images of other objects such as forks and knives nearby. Specifically, we first cluster images of similar categories (e.g., kitchen) using room labels predicted by a
Instruction rephrasing
We use a fill-in-the-blanks approach to replace noun-phrases in human annotated navigation instructions [5] by those in BnB captions (see Fig. 2). Concretely, we create more than 10K instruction templates containing 2-7 blanks, and fill the blanks with noun-phrases extracted from BnB captions. The noun-phrases matched to object categories from the Visual Genome [20] dataset are preferred during selection. This allows us to create VLN-like instructions with actionable verbs interspersed with room and object references for visual cues that are part of the BnB path (see Fig. 2).

Instruction generation
is a video captioning like model that takes in a sequence of images and generates an instruction corresponding to an agent’s path through an environment. To train this model, we adopt ViLBERT and train it to generate captions for single BnB image-caption pairs. Further, this model is fine-tuned on trajectories of the R2R dataset to generate instructions. Finally, we use this model to generate BnB PI pairs by producing an instruction for a concatenated image sequence from BnB (the path).

4. Airbert: A Pretrained VLN Model

In this section, we present Airbert, our multi-modal transformer pretrained on the BnB dataset with masking and shuffling losses. We first introduce the architecture of Airbert, and then describe datasets and pretext tasks in pretraining. Finally, we show how Airbert can be adapted to downstream VLN tasks.

4.1. ViLBERT-like Architecture

ViLBERT [24] is a multi-modal transformer extended from BERT [10] to learn joint visio-linguistic representations from image-caption pairs, as illustrated in Fig. 3.

Given an image-caption pair \((V, C)\), the model encodes the image as region features \([v_1, \ldots, v_V] \) via a pretrained Faster R-CNN [4], and embeds the text as a series of tokens: \([\text{CLS}], w_1, \ldots, w_T, \text{[SEP]}\), where \([\text{CLS}]\) and \([\text{SEP}]\) are special tokens added to the text. ViLBERT contains two separate transformers that encode \(V\) and \(C\) and it learns cross-modal interactions via co-attention [24].

We follow a similar strategy to encode path-instruction pairs (created in Sec. 3.2) that contain multiple images and captions \(\{(V_k, C_k)\}_{k=1}^K\). Here, each \(V_k\) is represented as visual regions \(v_k\) and \(C_k\) as word tokens \(w_k\). Respectively, the visual and text inputs to Airbert are:

\[
X_V = [(\text{IMG}), v_1, \ldots, v_K], \quad X_C = [(\text{CLS}), w_1, \ldots, w_K, \text{[SEP]}],
\]

where the \(\text{[IMG]}\) token is used to separate image region features taken at different locations.

Note that while our approach is not limited to a ViLBERT-like architecture, we choose ViLBERT for a fair comparison with previous work [28].

4.2. Datasets and Pretext Tasks for Pretraining

We use Conceptual Captions (ConCaps) [36] and BnB-PI in subsequent pretraining steps (see Fig. 3) to reduce the domain gap for downstream VLN tasks.

Previous multi-modal pretraining efforts [24, 28, 16] commonly use two self-supervised losses given image-
caption (IC) pairs or path-instruction (PI) pairs: (1) *Masking loss*: An input image region or word is randomly replaced by a [MASK] token. The output feature of this masked token is trained to predict the region label or the word given its multi-modal context. (2) *Pairing loss*: Given the output features of [IMG] and [CLS] tokens, a binary classifier is trained to predict whether the image (path) and caption (instruction) are paired.

The above two pretext tasks mainly focus on learning object-word associations instead of reasoning about the temporal order of paths and instructions. For example, if an image \( V_i \) appears before \( V_j \), then words from its caption \( C_i \) should appear before \( C_j \). In order to promote such a temporal reasoning ability, we propose an additional *shuffling loss* to enforce alignment between PI pairs.

Given an aligned PI pair \( X^+ = \{(V_k, C_k)\}_{k=1}^K \), we generate \( N \) negative pairs \( X^- = \{(V_k, C_l)\}, k \neq l \), by shuffling the composed images or the captions. We train our model to choose the aligned PI pair as compared to the shuffled negatives by minimizing the cross-entropy loss:

\[
L = -\log \frac{\exp(f(X^+))}{\exp(f(X^+)) + \sum_n \exp(f(X_n^+))}, \tag{3}
\]

where \( f(X) \) denotes the similarity score (logit) computed via Airbert for the PI pair \( X \).

### 4.3. Adaptations for Downstream VLN tasks

We consider two VLN tasks: goal-oriented navigation (R2R [5]) and object-oriented navigation (REVERIE [34]). Airbert can be readily integrated in discriminative and generative models for the above VLN tasks.

**Discriminative Model: Navigation as Path-Selection [28].** The navigation problem on the R2R dataset is formulated as a path selection task in [28]. Several candidate paths are generated via beam search from a navigation agent such as [40], and a discriminative model is trained to choose the best path among them. We fine-tune Airbert on the R2R dataset for path selection. A two-stage fine-tuning process is adopted: in the first phase, we use masking and shuffling losses on the PI pairs of the target VLN dataset in a manner similar to BnB PI pairs; in the second phase, we choose a positive candidate path as one that arrives within 3m of the goal, and contrast it against 3 negative candidate paths. We also compare multiple strategies to mine additional negative pairs (other than the 3 negative candidates), and in fact, empirically show that negatives created using shuffling outperform other options.

**Generative Model: Recurrent VLN-BERT [15].** The Recurrent VLN-BERT model adds recurrence to a state in the transformer to sequentially predict actions, achieving state-of-the-art performance on R2R and REVERIE tasks. We use our Airbert architecture as its backbone and apply it to the two tasks as follows. First, the language transformer encodes the instruction via self-attention. Then, the embedded [CLS] token in the instruction is used to track history and concatenated with visual tokens (observable navigable views or objects) in each action step. Self-attention and cross-attention on embedded instructions are employed to update the state and visual tokens and the attention score from the state token to visual tokens is used to decide the action at each step. We fine-tune the Recurrent VLN-BERT model with Airbert as the backbone in the same way as [15].

Additional details about the models and their implementation are provided in the Appendix B.

### 5. Experimental Results

We first perform ablation studies evaluating alternative ways to pretrain Airbert in Sec. 5.1. Then, we compare Airbert with state-of-the-art methods on R2R and REVERIE tasks in Sec. 5.2. Finally, in Sec. 5.3, we evaluate models in a more challenging setup: VLN few-shot learning where an agent is trained on examples taken from one/few houses.

**R2R Setup.** Most of our experiments are conducted on the R2R dataset [5], where we adopt standard splits and metrics defined by the task. We focus on success rate (SR), which is the ratio of predicted paths that stop within 3m of the goal.
Table 2: Comparison between various BnB PI pair creation strategies for pretraining. The first row denotes the use of image-caption pairs. All methods from the second row use masking and shuffling during pretraining. Cat: naive concatenation; Rep: instruction rephrasing; Gen: instruction generation; Merge: image merging; and Insert: captionless image insertion.

<table>
<thead>
<tr>
<th>BnB Mask</th>
<th>Speaker Shuf. Rank Shuf.</th>
<th>R2R Rank Shuf.</th>
<th>SR on Val Seen</th>
<th>SR on Val Unseen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>71.20</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>73.12</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>73.24</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>73.57</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>74.69</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>70.21</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>73.83</td>
</tr>
</tbody>
</table>

Table 3: Impact of shuffling during pretraining and fine-tuning. While additional data helps, we see that using the shuffling loss (abbreviated as Shuf.) consistently improves model performance. Row 1 corresponds to VLN-BERT [28].

<table>
<thead>
<tr>
<th>Fine-tuning Strategies</th>
<th>Additional Negatives</th>
<th>SR on Val Seen</th>
<th>SR on Val Unseen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 VLN-BERT [28]</td>
<td>0</td>
<td>70.20</td>
<td>59.26</td>
</tr>
<tr>
<td>2 (1) + Wrong trajectories</td>
<td>2</td>
<td>70.11</td>
<td>59.11</td>
</tr>
<tr>
<td>3 (1) + Highlight keywords</td>
<td>0</td>
<td>71.89</td>
<td>61.37</td>
</tr>
<tr>
<td>4 (1) + Hard negatives</td>
<td>2</td>
<td>71.89</td>
<td>61.63</td>
</tr>
<tr>
<td>5 (1) + Shuffling (Ours)</td>
<td>2</td>
<td>72.46</td>
<td><strong>61.98</strong></td>
</tr>
</tbody>
</table>

Table 4: Comparison between different strategies for fine-tuning a ViLBERT model on the R2R task. VLN-BERT [28] fine-tunes ViLBERT with a masking and ranking loss. Each row (described in the text) is an independent data augmentation and can be compared directly against the baseline (row 1).

Table 5: Accuracy of models attempting to pick the correct PI pair from a pool of correct + 10 negatives created by simple corruptions such as replacing or swapping noun phrases and switching directions (left with right). Random performance is \(\frac{1}{11}\) or 9.1%.

(RGSPL) is a path length weighted version.

### 5.1. Pretraining with BnB

We perform ablation studies on the impact of various methods for creating path-instruction pairs. We also present ablation studies that highlight the impact of using the shuffling loss during Airbert’s pretraining as well as fine-tuning stages. Throughout this section, our primary focus is on the SR on the unseen validation set and we compare our results against VLN-BERT [28], which achieves a SR of 59.26%.

#### 1. Impact of creating path-instruction pairs

Table 2 presents the performance of multiple ways of using the BnB dataset after ConCaps pretraining as illustrated in Fig. 3. In row 1, we show that directly using BnB IC pairs without any strategies to reduce domain gap improves performance over VLN-BERT by 3.2%. Even if we skip ConCaps pretraining, we achieve 60.54% outperforming 59.26% of VLN-BERT. It proves that our BnB dataset is more beneficial to VLN than the generic ConCaps dataset.

Naive concatenation (row 2) does only slightly better than using the IC pairs (row 1) as there are still domain shifts with respect to fluency of transitions and lack of visual context. Rows 3-6 show that each method mitigates domain-shift to some extent. Instruction rephrasing (row 3) performs better at improving instructions than instruction generation (row 4), possibly since the generator is unable to use the diverse vocabulary of the BnB captions. Inserting captionless images at random locations (row 6) reduces the domain-shift significantly and achieves the highest individual performance. Finally, a combination of instruction rephrasing, image merging and captionless insertion provides an overall 3.8% improvement over concatenation, and a large 7.2% improvement over VLN-BERT.

#### 2. Shuffling loss applied during pretraining

Table 3 demonstrates that shuffling is an effective strategy to train the model to reason about temporal order, and enforce alignment between PI pairs. Rows 3-5 show that shuffling is beneficial both during pretraining with BnB-PI data, or during fine-tuning with R2R data, and results in 2.3% and 0.4% improvements respectively. In combination with the Speaker dataset (paths from seen houses with generated instruction yielding 178K additional PI pairs [40]), we see that the shuffling loss provides 3.1% overall improvement (row 6).
3. Shuffling loss applied during fine-tuning. The final stage of model training on R2R involves fine-tuning to rank multiple candidate paths that form the path selection task. We compare various approaches to improve this fine-tuning procedure (results in Table 4). (1) In row 2, we explore the impact of using additional negative paths. Unsurprisingly, this does not improve performance. (2) Inspired by [12], we highlight keywords in the instruction using a part-of-speech tagger [17], and include an extra loss term that encourages the model to pay attention to their similarity scores (row 3). (3) Another alternative suggested by [12] involves masking keywords in the instruction and using VLP models to suggest replacements, resulting in hard negatives (row 4).

Hard negatives and highlighting keywords improve performance by 2.1-2.3%, but at the cost of extra parsers or VLP models. In contrast, shuffling visual paths to create two additional negatives results in highest improvement (row 5, +2.7% on val unseen) and appears to be a strong strategy to enforce temporal order reasoning, that neither requires external parsers nor additional VLP models.

4. Error analysis. We study the areas in which Airbert brings major improvements by analyzing scores for aligned PI pairs and simple corruptions that involve replacing noun phrases (e.g., bedroom by sofa), swapping noun phrases appearing within the instruction, or switching left and right directions (e.g., turn left/right or leftmost/rightmost chair). In particular, for every ground-truth aligned PI pair, we create 10 additional negatives by corrupting the instruction, and measure the accuracy of the model selecting the correct pair. Table 5 shows that Airbert with in-domain training and the shuffling loss achieves large improvements (> 8%) for corruptions involving replacement or swapping of noun phrases. On the other hand, distinguishing directions continues to be a challenging problem; but here as well we see Airbert outperform VLN-BERT by 4.5%.

5.2. Comparison against state-of-the-art

R2R. We first evaluate the discriminative model for the R2R task. Similar to VLN-BERT, we evaluate Airbert as an ensemble model created by a linear combination (chosen through grid search) of multiple model outputs (see Table 6). First, we see that Airbert alone (row 2) outperforms VLN-BERT (row 1) by 9.4% on the unseen environments.
and a strong ensemble of speaker and follower models [40] (row 3) by 0.7%. Ensembling Airbert results in a gain of 1.4% over the VLN-BERT ensemble (row 4 vs. 5).

We also obtain results on the test set by submitting our best method to the R2R leaderboard. As seen from Table 8, our method of ensembling Airbert, speaker, and follower (similar to VLN-BERT with speaker and follower [10]) achieves the highest success rate at 77% and is ranked first as of the submission deadline. Both VLN-BERT and Airbert use 30 candidate trajectories sampled by beam search with EnvDrop [40], inducing the same path length (PL) for the three methods. As the SPL metric on the leaderboard takes into account the total path length over the 30 trajectories, the SPL is very low and similar across the approaches. Airbert also benefits generative models for the R2R task. The results are presented in the Appendix C.

**REVERIE.** Table 7 presents results for the REVERIE dataset. The last four rows in the table use Recurrent VLN-BERT [15] with different backbones or parameter initialization. The OSCAR and ViLBERT backbones are pretrained on out-of-domain image-caption pairs. As compared to OSCAR, we observe slight improvements using the ViLBERT backbone for the REVERIE task. VLN-BERT shares the same architecture as ViLBERT, but is pretrained on the R2R dataset, resulting in performance improvement on the unseen environments. Our pretrained Airbert achieves significantly better performance than VLN-BERT, with over 2.4% gain on navigation SR and 1.8% gain on RGS in unseen environments (val unseen). Without any special adaptation, we see that Airbert brings benefits from pretraining on the BnB dataset. We also achieve the state-of-the-art performance on the REVERIE test set by the time of submission, surpassing previous works by a large margin.

### 5.3. Training a navigation agent on few houses

We hypothesize that in-domain pretraining, especially one that leverages proposed PI pair generation methods, can achieve superior performance while requiring less training data. To evaluate this, we propose a novel few shot evaluation paradigm for VLN: models are allowed to fine-tune on samples (PI pairs) from one (or few) environments. Few-shot learning for VLN is particularly interesting as visual appearance of houses may differ vastly across geographies, and while training data is hard to obtain, pretraining data like BnB may be readily available.

**One/few shot tasks.** We considered two types of setups: (1) learning from a single environment, which we refer as one-shot learning; and (2) learning from 6 environments (representing 10% of the total training size). For both cases, we randomly sample 5 sets of environments, and report average results (standard deviations in the Appendix C). As the number of paths in an environment may have a large impact on performance, we exclude 17 of 61 environments with less than 80 paths.

**Results.** We adopt VLN-BERT, pretrained on ConCaps, as a baseline for few-shot tasks. Recall that fine-tuning VLN-BERT and Airbert on R2R relies on candidate paths drawn from an existing model (EnvDrop [40]). However, as this would lead to unfair comparisons (EnvDrop is trained on the full dataset), candidate paths are sampled as the shortest path between two random positions.

Table 9 shows that Airbert largely outperforms VLN-BERT on the unseen validation set: 27.6% with 1 house and 22% with 6 houses. Airbert fine-tuned on 6 houses is almost as good as VLN-BERT on the entire training set. The last two rows of the table shows that using random paths does not lead to a large performance drop for both models and is a testament to the power of pretrained networks.

### 6. Conclusion

We introduced BnB, a large-scale, in-domain, image-text dataset from houses listed on online rental marketplaces and showed how domain gaps between BnB image-caption pairs and VLN tasks can be mitigated through the creation of path-instruction pairs. We also proposed shuffling, as a means to improve an agent’s reasoning about temporal order. Our pretrained model Airbert, achieved state-of-the-art on R2R through the discriminative path-selection setting, and REVERIE through a generative setting. We also demonstrated large performance improvements when applying our model to a challenging one/few-shot VLN setup, highlighting the impact of good pretraining in VLN tasks.

**Acknowledgments.** This work was granted access to the HPC resources of IDRIS under the allocation 101002 made by GENCI. It was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute) and by Louis Vuitton ENS Chair on Artificial Intelligence.
References


[22] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. EMNLP, 2019.


[34] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments. In CVPR, 2020. 2, 3, 5, 6, 7
[38] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In ICLR, 2019. 1
[46] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In CVPR, 2016. 2
Appendix

In this supplementary material, we present additional details, statistics and examples for the BnB dataset; we discuss implementation details for the models used in our work; and present qualitative results as well as the detailed results for the new few-shot learning paradigm.

A. BnB dataset

This section presents additional details for our Bed-and-Breakfast (BnB) dataset. We start by a short discussion of image-caption pairs (BnB IC) collected from an online rental marketplaces and their statistics. Subsequently, we present how a combinatorially large number of path-instruction pairs (BnB PI) can be created automatically. We end this section with multiple examples of BnB PI pairs generated via the concatenation and domain-shift reduction (e.g. rephrasing, captionless insertion) strategies.

A.1. Filtering image-caption pairs: Outdoor images

Images of outdoor scenes are almost never seen in the environments used in downstream VLN tasks. In fact, not only are the images out-of-domain (such images are rarely seen in the VLN environments), their captions are often irrelevant to a VLN task. In order to alleviate the impact of such noisy images and captions, we discard outdoor images from the pretraining process. Figure 4 illustrates several examples of misleading outdoor image-caption pairs. Captions as written by the host are presented in the label below the image. The caption for the image in Figure 4a refers to a “bedroom”, however, the image does not show a bedroom. Similarly, the image-caption pair in the Figure 4b talks about activities or festivals that take place in the neighborhood of the listing, however, they are not relevant for solving indoor navigation tasks. Finally, Figure 4c shows an outdoor scene with several birds along with a noisy caption that is not directly related to the image content, but the emotion that the image may evoke.

A.2. Dataset details and Statistics

BnB image-caption pairs. We collect BnB IC pairs from 150K listings on Airbnb resulting in 713K image-caption pairs and 676K images without captions. In Figure 5, we present some key statistics about this data. Figure 5a presents a histogram of the number of images found in each listing. While most listings have less than 20 images, this is still a sufficiently large and diverse in-domain distribution. In Figure 5b, we summarize the rooms depicted in the images through predicted category labels obtained using a CNN trained on the Places365 dataset [50]. These category labels are used as part of our proposed extensions such as image merging.

Creating BnB path-instruction pretraining samples. We create the BnB PI pairs on-the-fly during training to mimic the agent’s visual trajectory and a corresponding instruction through an environment. Each sample in a batch is created by randomly sampling a listing without replacement during an epoch (one epoch consists of one PI pair from each listing). Then, the number of IC pairs K that form the PI pair are chosen (as an integer) from a uniform distribution, $K \sim U[4, 7]$. We sample $N \sim U[2, K]$ IC pairs that have a non-empty caption and the remainder $K - N$ images are chosen from the set of captionless images. Any image in the path may include additional visual context (from the same room) via the image merging strategy. Similarly, the instruction rephrasing strategy may be employed by using existing R2R instruction templates and filling them with noun phrases extracted from the image captions.

The above procedure results in creating one correctly aligned (positive) PI pair, $(X^+ , X^-)$ in the main paper). To employ the shuffling loss for each sample, we create 9 additional negatives $(X^-_{i})$ in the paper) by shuffling either the sequence of images or captions, ensuring that the post-shuffling order does not align with the positive pair.

Statistics for BnB PI pairs. Due to the large number of possible combinations, we can (theoretically) create 200 billion path-instruction pairs, using the simple concatenation strategy. This number grows to over 300 quadrillion when considering additional visual context augmentations and fluent instructions.

For instruction rephrasing, we create 11,626 fill-in-the-blank templates from the R2R training set. Figure 5c shows the distribution of the number of blanks in the templates – most instruction templates have 2-7 blanks into which we insert noun phrases from the BnB captions.

While we are unable to generate the entire BnB PI dataset for computing statistics, we generate 50K PI pairs as a representative sample. Figure 5d presents the distribution of instruction lengths (number of words) for different datasets. We see that the captions in BnB IC pairs are much shorter than typical instructions in R2R and REVERIE, while our automatically created instructions in BnB PI pairs exhibit a high level of similarity in terms of their length.

A.3. Examples of BnB PI Pairs

Figure 6 presents generated BnB PI pairs using various strategies proposed in our work, including naive concatenation, instruction rephrasing, instruction generation, image merging and captionless image insertion.

Among the methods to create an instruction, simple concatenation lacks action verbs between sentences for fluent transition leading to a domain shift from real instructions. Instruction rephrasing selects noun phrases from BnB image descriptions and inserts them into real instruction templates, providing a natural feel to the created instruction.
Finally, while the learning approach of instruction generation (recall, this is learned on downstream VLN dataset) produces fluent sentences, it is unable to leverage the diverse captions of BnB images due to the limited vocabulary stemming from the downstream VLN dataset. For example, the generated instruction in Figure 6c does not contain noun phrases related to images in the path. Better caption generation models such as Pointer network [43] may help avoid such problems, however are left for future work.

Among augmentations for path generation, we can see that image merging helps to expand relevant visual context from single images to semi-panoramic views, see the bedroom in Figure 6a or the kitchen in Figure 6b. Captionless image insertion also improves the path diversity by mimicking unmentioned viewpoints in the instruction (indicated by images with a dotted border).
B. Implementation details

We present the implementation details for learning Airbert via pretraining using BnB, and subsequent fine-tuning in both discriminative or generative settings.

B.1. Airbert Pretraining

Airbert’s architecture is the same as VLN-BERT (see Figure 7a where the number of layers \(L_1 = L_2 = 6\)). The feature vector \(v_k^i\) (corresponding to \(i\)th image region of the \(k\)th image) is composed of three terms: the first term is the visual feature extracted by the Bottom-Up Top-Down attention model [4]; the second term encodes the location of the region in the image as MLP(\(l_k^i\)), where \(l_k^i\) is the 5-dim location vector of the given image region defined as the top corner \((x, y)\), the width, height and area; and the last term \(\text{Emb}(k)\) encodes the position, where \(\text{Emb}\) is an embedding layer for the image order.

We use 8 V100 SXM2 GPUs (32 GB each) for pretraining Airbert. The model is trained for 15 epochs with a batch size of 64 and learning rate of \(4 \times 10^{-5}\). Each epoch consists of one randomly sampled PI pair from 95% of the listings, while the remaining 5% are used for validation and preventing overfitting.

B.2. Fine-tuning in Discriminative Setting

In the discriminative setting, R2R navigation is formulated as a path selection problem given the instruction. The pretrained Airbert model can be directly fine-tuned without any modifications to the architecture to predict the path-instruction alignment (or compatibility) score as shown in Figure 7a.

We follow the same fine-tuning setup as VLN-BERT [28] to allow for a fair comparison. We use the Adam optimizer with a learning rate of \(4 \times 10^{-5}\). The optimizer is controlled by a learning rate scheduler with a linear warmup and cooldown. We fine-tune Airbert for 30 epochs with a batch size of 64. Samples from the R2R training set are used
(a) Adapting Airbert to a discriminative setting to predict path-instruction alignment score, similar to [28].

(b) Adapting Airbert to a generative setting based on the Recurrent VLN-BERT [15].

Figure 7: The adapted Airbert model in both discriminative and generative settings for downstream VLN tasks.
### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validation Seen</th>
<th>Validation Unseen</th>
<th>Test Unseen</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TL  NE  SR  SPL</td>
<td>TL  NE  SR  SPL</td>
<td>TL  NE  SR  SPL</td>
</tr>
<tr>
<td>PRESS [22]</td>
<td>10.57  4.39  58  55</td>
<td>10.36  5.28  49  45</td>
<td>10.77  5.49  49  45</td>
</tr>
<tr>
<td>EnvDrop [40]</td>
<td>11.00  3.99  62  59</td>
<td>10.70  5.22  52  48</td>
<td>11.66  5.23  51  47</td>
</tr>
<tr>
<td>Rec (no init. OSCAR) [15]</td>
<td>9.78  3.92  62  59</td>
<td>10.31  5.10  50  46</td>
<td>11.15  5.45  51  47</td>
</tr>
<tr>
<td>Rec (OSCAR) [15]</td>
<td>10.79  3.11  71  67</td>
<td>11.86  4.29  59  53</td>
<td>12.34  4.59  57  53</td>
</tr>
<tr>
<td>Rec (PREVALENT) [15]</td>
<td>11.13  2.90  72  68</td>
<td>12.01  3.93  63  57</td>
<td>12.35  4.09  63  57</td>
</tr>
<tr>
<td>Rec (ViLBERT)</td>
<td>11.16  2.54  75  71</td>
<td>12.44  4.20  60  54</td>
<td>-  -  -  -</td>
</tr>
<tr>
<td>Rec (VLN-BERT)</td>
<td>10.95  3.37  68  64</td>
<td>11.33  4.19  60  55</td>
<td>-  -  -  -</td>
</tr>
<tr>
<td>Rec (Airbert)</td>
<td>11.09  2.68  75  70</td>
<td>11.78  4.01  62  56</td>
<td>12.41  4.13  62  57</td>
</tr>
</tbody>
</table>

Table 10: Navigation performance of different generative models on the R2R dataset.

Replacing OSCAR’s single BERT-like architecture with the ViLBERT architecture slightly improves the performance (similar to our results on the REVERIE dataset presented in the main paper). The VLN-BERT model further fine-tunes ViLBERT on the R2R dataset (with the masking loss). This is beneficial to the navigation performance on the unseen environments validation set.

Our Airbert initialization achieves substantial performance improvement as compared to the OSCAR and VLN-BERT backbones on unseen environments, while achieving comparable performance with the PREVALENT initialization.

### C.2. Qualitative results

We visualize the predicted paths from VLN-BERT and Airbert models. In the following figures, ● is the starting viewpoint of the agent, ■ denotes viewpoints in the ground-truth path, ■ for VLN-BERT and ■ for Airbert. Arrows indicate the navigation direction.

**New houses.** In Figure 8, we compare predicted paths from VLN-BERT and Airbert in new houses beyond the training environments. Benefiting from BnB dataset that provides diverse visual environments in pretraining, our Airbert model generalizes better to recognize different room types in new houses (see Figure 8a-8d), and performs better on significantly different environments such as “lounge chairs” and “cabinet” besides the last step by learning from the shuffling task. Similarly, in Figure 10c, we see that the VLN-BERT agent stops at the wrong stairs, while Airbert considers intermediate steps such as “hallway” and “wooden doors”, and ends within the acceptable range of 3m from the goal.

**New objects.** Airbert also improves the understanding of new objects in home environments, e.g. through noun phrases related to household objects. As shown in Figure 9, it is successful at following instructions containing noun phrases that rarely occur or are even unseen on the training set, while the VLN-BERT model that is trained on a large image-caption corpus not pertaining to houses fails.

Failure cases. Figure 11 presents some failure cases for both VLN-BERT and Airbert. It reveals that current models still struggle to deal with relationships such as “between” (Figure 11a), or directional instructions such as “on the left” (Figure 11b). Similar failures are also highlighted by Table 5 of the main paper where we show that models fail to choose the correct instruction when a direction keyword (left/right) is switched.

### C.3. Few-shot Learning on VLN

As mentioned in the main paper, we present complete results for the few-shot learning evaluation, along with standard deviations in Table 11. While the performance on the seen validation houses fluctuates a lot (also due to changing the environment in the seen validation set), unseen validation is very stable. Recall that VLN-BERT achieves an unseen validation performance of 27% and 37% with 1 and 6 training environments respectively. On the other hand, Airbert achieves a superior 49.5% and 58% – an absolute improvement of ~22% in both cases.
(a) **R2R ✔:** Walk over the kitchen counter, turn left, walk ahead till wall, turn right, walk to the closet room, wait at front.

(b) **REVERIE ✔:** Walk past the kitchen and enter the hallway. Turn right at the artwork and wait by the closet.

(c) **R2R ✔:** Walk forward to the sitting area to the right of the stairs. Walk to the wall of windows and take a right into the recreation room and stop before you reach the pool table.

(d) **R2R ✔:** Go between the counters, turn left, turn right, and stop before the display and dining room.

(e) **R2R ✔:** Turn right and head towards the end. Once you reach the end make a right and stop.

(f) **R2R ✔:** Walk straight out the door in front of you and follow the red carpet. Keep going through the room with the ropes and stop when you enter the next room with ropes.

Figure 8: When navigating in new houses, our Airbert model not only successfully recognizes the closet room in (a) and (b), pool table (c), living room (d), but also generalizes better to challenging environments, such as the church (e) and castle (f).
(a) **R2R ✔**: Walk up the stairs and take a right. Walk into the bedroom and take a left. Take another left at the night stand and walk out of the bedroom. Wait by the toilet in the second door on the right.

(b) **R2R ✔**: Go straight past the table and chairs then turn left and continue to go past the table and chairs. Wait near the white antique furniture with the two chairs on each side.

(c) **REVERIE ✔**: Walk past the pool table and towards the TV on the far side of the room and grab the coffee table that is located in front of the couch.

(d) **REVERIE ✔**: Please go to the pantry room with the two large freezers and kitchen appliances on the large table and reset the flipped breaker in the breaker panel box to the right of the freezers.

Figure 9: The Airbert model outperforms VLN-BERT to recognize rare or even unseen objects in training set. (a) Rare object “night stand”; (b) unseen object “antique furniture”; (c) rare object “pool table”; and (d) unseen object “freezer”.

<table>
<thead>
<tr>
<th># Env.</th>
<th>Traj.</th>
<th>Val Seen SR</th>
<th>Val Unseen SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PL</td>
<td>NE</td>
</tr>
<tr>
<td>1</td>
<td>Rand</td>
<td>10.97</td>
<td>5.36</td>
</tr>
<tr>
<td>6</td>
<td>Rand</td>
<td>9.84</td>
<td>5.49</td>
</tr>
<tr>
<td>61</td>
<td>Rand</td>
<td>10.91</td>
<td>4.87</td>
</tr>
<tr>
<td>61</td>
<td>[40]</td>
<td>10.59</td>
<td>3.21</td>
</tr>
</tbody>
</table>

Table 11: Performance of Airbert on R2R few-shot evaluation. During training, only a subset of the Matterport [7] environments are accessible.
(a) R2R ✔: Walk from dining room to living room turning slightly right before lounge chairs, walk straight following cabinet. Turn slight right and stop at stairs.

(b) R2R ✔: Walk on into the kitchen and turn to the right. Walk past the staircase, behind the chairs. Walk to the right of the pillar. Stop and wait by the footstool.

(c) R2R ✔: Walk out of the hallway and turn left. Walk down the steps and through the wooden doors. Walk down the steps and stop.

(d) R2R ✔: Go straight passed the coffee table turn left and go through the left door to the stairs. Stop in front of the stairs.

Figure 10: Examples in similar environments and instructions to the training set. The improvements of Airbert model can be contributed to the shuffling loss in pretraining.
(a) **R2R**: Walk between the two kitchen islands and then turn right. Pass through the stone archway and stop just after you pass through it. Wait there.

(b) **R2R**: Exit the bathroom and go down the stairs. Enter the last doorway on the left and stop just before stepping on the rug.

(c) **REVERIE**: Go to level 3 bathroom in the first bedroom left of the stairs and grab the mirror on the wall

(d) **REVERIE**: Go to the lounge on this level and polish the black leather armchair in the corner

Figure 11: Failure cases for both VLN-BERT and Airbert models.