Improving Vision-and-Language Navigation with Image-Text Pairs from the Web

Arjun Majumdar[1], Ayush Shrivastava[1], Stefan Lee[3],
Peter Anderson[1⋆], Devi Parikh[1,2], and Dhruv Batra[1,2]

[1]Georgia Institute of Technology   [2]Facebook AI Research   [3]Oregon State University

Abstract. Following a navigation instruction such as *'Walk down the stairs and stop at the brown sofa'* requires embodied AI agents to ground referenced scene elements referenced (e.g. *'stairs'*) to visual content in the environment (pixels corresponding to *'stairs'*). We ask the following question – can we leverage abundant 'disembodied' web-scraped vision-and-language corpora (e.g. Conceptual Captions) to learn the visual groundings that improve performance on a relatively data-starved embodied perception task (Vision-and-Language Navigation)? Specifically, we develop VLN-BERT, a visiolinguistic transformer-based model for scoring the compatibility between an instruction (*'...stop at the brown sofa'*) and a trajectory of panoramic RGB images captured by the agent. We demonstrate that pretraining VLN-BERT on image-text pairs from the web before fine-tuning on embodied path-instruction data significantly improves performance on VLN – outperforming prior state-of-the-art in the fully-observed setting by 4 absolute percentage points on success rate. Ablations of our pretraining curriculum show each stage to be impactful – with their combination resulting in further gains.

Keywords: vision-and-language navigation, transfer learning, embodied AI

1   Introduction

Consider the navigation instruction in Figure 1, *'Walk through the bedroom and out of the door into the hallway. Walk down the hall along the banister rail through the open door. Continue into the bedroom with a round mirror on the wall and butterfly sculpture.'* In vision-and-language navigation (VLN) [4], agents must interpret such instructions to navigate through photo-realistic environments. In this instance, the agent needs to exit the bedroom, walk past something called a *'banister rail'* and find the bedroom containing a *'round mirror'* and *'butterfly sculpture.'* But what if the agent has never seen a butterfly before, let alone a sculpture of one? To solve this task, an agent needs to determine if the visual evidence along a path matches the descriptions provided in the instructions. As such, the ability to ground references to objects and scene
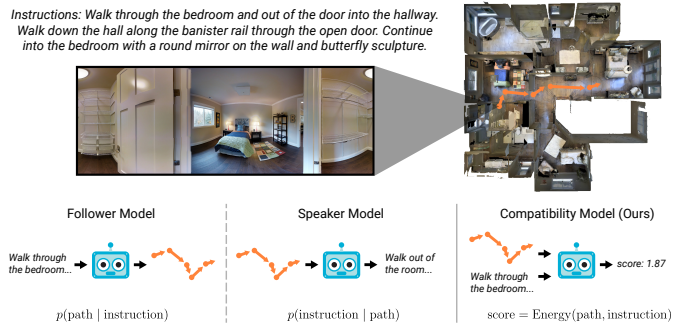

⋆ Now at Google.

Fig. 1. We propose a compatibility model (right) for path selection in vision-and-language navigation (VLN). In contrast to the follower (left) and speaker (center) models that have typically been used in prior work, our model takes a path and instruction pair as input and produces a score that reflects their alignment. Based on this model we describe a training curriculum that leverages internet data in the form of image-caption pairs to improve performance on VLN.

elements like *'butterfly sculpture'* and *'banister rail'* is central to success. Existing work has focused on learning this grounding solely from a task-specific training dataset of path-instruction pairs [3, 8, 13, 20, 21, 28, 30, 31] – which are expensive, laborious, and time-consuming to collect at scale and thus tend to be relatively small (e.g. the VLN dataset contains around 14k path-instruction pairs for training). As an alternative, we propose learning visual grounding from freely-available internet data, such as the web images with alt-text captured in the Conceptual Captions dataset [24], containing around 3.3M image-text pairs.

Conceptually, transfer learning from large-scale web data to embodied AI tasks such as VLN is an attractive alternative to collecting more data. Empirically, however, the effectiveness of this strategy remains open to question. Unlike web images, which are highly-curated and with clear aesthetic biases, embodied data contains content and viewpoints that are not widely published online. As shown in Figure 2, an embodied agent may perceive doors via a close-up view of a door frame rather than as a carefully composed image of a (typically closed) door. In VLN, image framing is a consequence of the agent's position rather than an aesthetic choice made by a photographer. Consequently, in this paper we investigate this question – to what degree can visual grounding learned on static web images be transferred to the embodied VLN task? That is, can 'disembodied' web data be used to improve visual grounding for embodied agents?

To answer this question, we introduce VLN-BERT, a joint visiolinguistic transformer-based compatibility model for scoring the alignment between an instruction and an agent's observations along a trajectory. We structure VLN-BERT to enable straight-forward transfer learning from a model from prior work on general visiolinguistic representation learning [18], and explore a training curriculum that incorporates both large-scale internet data and embodied path-instruction pairs. VLN-BERT is sequentially trained using 1) language-only data

Fig. 2. Images from the Conceptual Captions (CC) [24] (top) and Matterport3D (MP3D) [5] (bottom) datasets illustrate the differences between the two domains. Images from CC are typically well-lit, well-composed and aesthetically pleasing, while for MP3D images (used in VLN) the framing depends on the position of the agent (e.g. a couch (left) in CC is typically viewed head-on, whereas in MP3D they may be hidden to the side as an agent navigates past them).

(Wikipedia and BooksCorpus [34] as in BERT [7]), 2) web image-text pairs (Conceptual Captions [24] as in ViLBERT [18]), and 3) path-instruction pairs from the VLN dataset [4]. Following this protocol the model progressively learns to represent language, then to ground visual concepts, and finally to ground visual concepts alongside action descriptions. We evaluate VLN-BERT on a path selection task in VLN, demonstrating that this training procedure leads to significant gains over prior work (4 absolute percentage points on leaderboard success rate).

Contributions. Concretely, we make the following main contributions:

– We develop VLN-BERT, a visiolinguistic transformer-based model for scoring path-instruction pairs. We show that VLN-BERT outperforms strong single-model baselines from prior work on the path selection task – increasing success rate (SR) by 4.6 absolute percentage points.
– We demonstrate that in an ensemble of diverse models VLN-BERT improves SR by 3.0 absolute percentage points on "unseen" validation, leading to a SR of 73% on the VLN leaderboard (4 absolute percentage points higher than previously published work)[1].
– We ablate the proposed training curriculum, and find that each stage contributes significantly to the final outcome, with a cumulative benefit that is greater than the sum of the individual effects. Notably, we find that pretraining on image-text pairs from the web provides a significant boost in path selection performance – improving SR by 9.2 absolute percentage points.

2 Related Work

Path Selection in VLN. In VLN [4], an agent is required to follow a navigation instruction from a start location to a goal. While most existing works

---

[1] evalai.cloudcv.org/web/challenges/challenge-page/97/leaderboard/270

focus on the setting in which the test environments are previously unseen, many also consider the scenario in which the test environment is previously explored and stored in memory (i.e., fully observable). In this setting, a high-probability path is typically generated by performing beam search through the environment and ranking paths according to either: (1) their probability under a 'follower' model [3, 8, 13, 20, 21, 30, 31], as in Figure 1 (left), or (2) by how well they explain the instruction according to a 'speaker' (instruction generation) model [8, 28], as in Figure 1 (center). In contrast, we use beam search with an existing agent model [28] to generate a set of candidate paths, which we then evaluate using our discriminative path-instruction compatibility model, as in Figure 1 (right).

Data Augmentation and Auxiliary Tasks in VLN. To compensate for the small size of existing VLN datasets, previous works have investigated various data augmentation strategies and auxiliary tasks. Many papers report results trained on augmented data including instructions synthesized by a speaker model [8, 20, 21, 28]. Tan et al. [28] use environmental dropout to mimic additional training environments to improve generalization. Li et al. [17] incorporate language-only pretraining using a BERT model. Several existing papers [11, 30] and one concurrent hitherto-unpublished work [10] consider path-instruction compatibility as an auxiliary loss function or reward for VLN agents. We focus on path-instruction compatibility in the context of transfer learning from large-scale internet data, which has not been previously explored.

Vision-and-Language Pretraining. There has been significant recent progress towards learning transferable joint representations of images and text [15, 16, 18, 26, 27, 32]. Using BERT-like [7] self-supervised objectives and Transformer [29] architectures, these models have achieved state-of-the-art results on multiple vision-and-language tasks by pretraining on aligned image-and-text data collected from the web [24] and transferring the base architecture to other tasks such as VQA [9], referring expressions [12], and caption-based image retrieval [6]. However, these tasks are all based on single images. The extent to which these pretrained models can generalize from human-composed and curated internet images to embodied AI tasks has not been investigated. In this work we propose a training curriculum to handle potential domain-shift and augment a previous model architecture to process panoramic image sequences, extending the progress in vision-and-language to vision-and-language navigation (VLN).

3   Preliminaries: Self-Supervised Learning from the Web

Recent works have demonstrated that high-capacity models trained under self-supervised objectives on large-scale web data can learn strong, generalizable representations for both language and images [7, 15, 16, 18, 26, 27, 32]. We build upon these works as a basis for transfer and describe them briefly here.

Language Modeling with BERT. The BERT [7] model is a large transformer-based [29] architecture for language modeling. The input to the model is sequences of tokenized words augmented with positional embeddings and outputs a representation for each. For example, a two sentence input could be written as

$$\texttt{<CLS>}\ w_1^{(1)}, \ldots, w_{L_1}^{(1)}\ \texttt{<SEP>}\ w_1^{(2)}, \ldots, w_{L_2}^{(2)}\ \texttt{<SEP>} \tag{1}$$

where CLS, and SEP are special tokens. To train this approach, [7] introduce two self-supervised objectives – *masked language modelling* and *next sentence prediction*. Given two input sentences from a text corpus, the masked language modelling objective masks out some percentage of tokens and tasks the model to predict their values given the remaining tokens as context. The next sentence prediction objective requires the model to predict whether the two sentences follow each other in the original corpus or not. BERT is then trained under these objectives on large language corpuses from the web (Wikipedia and BooksCorpus [34]). This model forms the basis for both our approach and the visiolinguistic representation learning discussed next.

Visiolinguistic Representations Learning with ViLBERT. Extending BERT, ViLBERT [18] (and a number of similar approaches [15, 16, 18, 26, 27, 32]) focuses on learning joint visiolinguistic representations from paired image-text data, specifically web images and their associated alt-text collected in the Conceptual Captions dataset [24]. ViLBERT is composed of two BERT-like processing streams that operate on visual and textual inputs, respectively. The input to the visual stream is composed of image regions (generated by an object detector [2, 22] pretrained on Visual Genome [14]) that act as "words" in the visual domain. Concretely, given a single image I consisting of a set of image regions $\{v_1, \ldots, v_k\}$ and a text sequence (i.e. a caption) $w_1, \ldots, w_L$, we can write the input to ViLBERT as the sequence

$$\texttt{<IMG>}\ v_1, \ldots, v_k\ \texttt{<CLS>}\ w_1, \ldots, w_L\ \texttt{<SEP>} \tag{2}$$

where IMG, CLS, and SEP are special tokens marking the different modality subsequences. The two streams are connected using co-attention [19] transformer layers, which attend from the visual stream over language stream and vice versa. Notably, the language stream of ViLBERT is designed to mirror BERT such that it can be initialized by a pretrained BERT model. After processing, the model produces a contextualized output representation for each input token.

In analogy to the training objectives in BERT, ViLBERT introduces the *masked multimodal modelling* and *multimodal alignment tasks*. In the first, a random subset of language tokens and image regions are masked and must be predicted given the remaining context. For image regions, this amounts to predicting a distribution over object classes present in the masked region. Masked text tokens are handled as in BERT. The multimodal alignment objective trains the model to determine if an image-text pair matches, i.e. if the text describes the image content. Individual token outputs are used to predict masked inputs in the masking objective, and the IMG and CLS tokens are used for the image-caption alignment objective. We build upon this model extensively in this work.
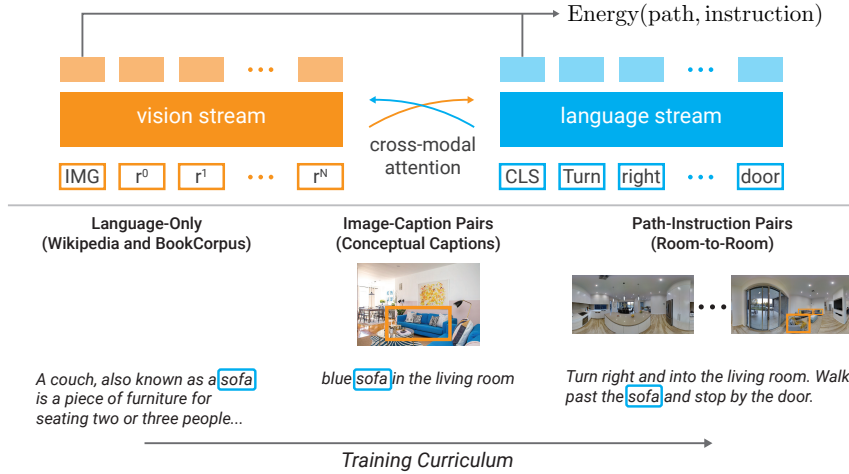
Fig. 3. We propose VLN-BERT (top), a visiolinguistic transformer-based model similar to the model from [18], to process image regions from a sequence of panoramas and words from an instruction. We demonstrate that with the proposed training curriculum (bottom) visual grounding learned from image-text pairs from the web (center) can be transferred to significantly improve performance in VLN.

4   Approach

4.1   Vision-and-Language Navigation as Path Selection

In the Vision-and-Language Navigation (VLN) [4] task, agents are placed in an environment specified as a navigation-graph $G = \{\mathcal{V}, \mathcal{E}\}$. Nodes $v \in \mathcal{V}$ represent different positions within the environment, and are represented by 360-degree panoramas taken at that viewpoint. Meanwhile, edges delineate navigable paths between panorama positions. The agent is provided with a navigation instruction x that describes the shortest-path between a starting position v_s and goal position v_g (as illustrated in the bottom right of Figure 3). Agents are considered to have succeeded if they traverse a path $\tau = [v_s, v_1, v_2, \ldots, v_N]$ with a final position v_N that is within 3m of the goal v_g.

Much of the work in VLN focuses on this problem as an exploration task in new environments; however, many practical deployments of robotic agents would be long-term in relatively fixed environments (e.g. an assistant operating in a single home). In this paper, we consider the setting in which the environment is previously explored with the navigation-graph and panoramas are stored in the agent's memory. This setting has been studied in prior work [8,13,20,28,30] and is operationalized by providing the agent unrestricted access to the Matterport3D Simulator [4] during inference such that it can consider arbitrarily many valid paths originating from the starting position v_s, before selecting one to follow.

In this setting, the navigation task becomes one of identifying the path best aligned with the instructions. Concretely, given a set of valid paths $\mathcal{T}$ with the

same starting position v_s and an instruction x, the problem of navigation is to identify a trajectory τ^* such that

$$\tau^* = \underset{\tau \in \mathcal{T}}{\operatorname{argmax}} \quad f(\tau, x) \tag{3}$$

for some compatibility function f that determines if the trajectory follows the instruction and terminates near the goal. The two major challenges are how to learn a compatibility function f and how to efficiently search through the large set of possible paths. Given that our focus is on transfer learning, we address the first challenge within a simple path selection setting. Specifically, we consider a small set of paths $\mathcal{T}' = \{\tau^1, \tau^2, \ldots, \tau^M\}$ for each instruction, which are generated using beam-search with a greedy instruction-following agent [28], and task f with selecting the path that best aligns with the instruction from this set. Future work might explore how f could be further used as a heuristic to efficiently search through the larger, exhaustive set of candidate paths $\mathcal{T}$.

4.2   Modeling Instruction-Path Compatibility

To formalize the task, we consider a function f that maps a trajectory τ and an instruction x to compatibility score $f(\cdot, \cdot)$. We model $f(\tau, x)$ as a visiolinguistic transformer-based model denoted as VLN-BERT. The architecture of VLN-BERT is structural similar to ViLBERT [18]; this is by design because it enables straight-forward transfer of visual grounding learned from large-scale web data. Specifically, we make a number of VLN-specific adaptations to ViLBERT, but they are all structured as augmentations (adding modules) rather than ablations (removing existing network components) so that pretrained weights can be transferred to initialize large portions of the model.

Representing Trajectories and Instructions. Predicting path-instruction compatibility requires jointly reasoning over a sequence of observations and a sequence of instruction words. As in prior work [8], a trajectory is represented as a sequence of panoramic images (as in Figure 3 bottom right) with positional information – i.e. $\tau = [(I_1, p_1), \ldots, (I_N, p_N)]$ where (I_i) are panoramas and (p_i) are poses. Further, we represent each panorama I_i as a set of image regions $\{r_1^{(i)}, \ldots, r_K^{(i)}\}$. Let an instruction x be a sequence of tokens $w_1, \ldots, w_L$. We can thus write a path-instruction pair for VLN-BERT as the input sequence

$$\texttt{<IMG>} \ r_1^{(1)}, \ldots, r_K^{(1)}, \ldots, \texttt{<IMG>} \ r_1^{(N)}, \ldots, r_K^{(N)}, \ \texttt{<CLS>} \ w_1, \ldots, w_L \ \texttt{<SEP>} \tag{4}$$

where `IMG`, `CLS`, and `SEP` are special tokens as before.

The transformer models on which VLN-BERT (as well as BERT and ViLBERT) is based are inherently invariant to sequence order – only representing interactions between inputs as a function of their values. The common practice to introduce this information is to add positional embeddings to the input token representations. For language, this is straight-forward and amounts to an index-in-sequence encoding. Panorama trajectories on the other have significantly more
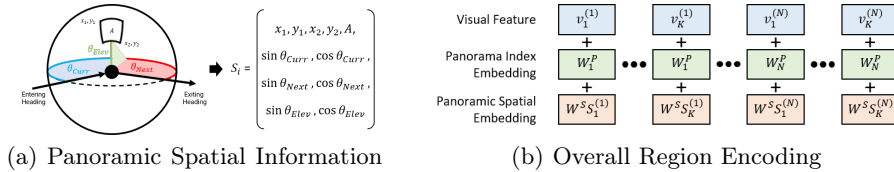
(a) Panoramic Spatial Information   (b) Overall Region Encoding

Fig. 4. We encode spatial information for each region to include not only the region position, but also its relation to the trajectory path (a). We form overall region encoding by summing visual features, an embedding indicating the index of the source panorama in the trajectory, and an embedding of panoramic spatial information (b).

complex relationships. While the panoramas themselves are a sequence, there are also geometric relationships between them (e.g. two panoramas being 1.2 meters apart at 10 degrees off north). Further, each individual image region not only has a position in the image (as modelled in ViLBERT) but also an angle relative to the heading of an agent as it traverses the trajectory. These are important considerations for language-guided navigation – after all, something on your left going one way is on your right if you go in the opposite direction. Being able to reason about the order of panoramas and the relative heading of image content is integral for following instructions like *'Go down the hallway on the right then stop when you see a table on your left.'*.

To address this, as visualized in Figure 4(a), we encode the spatial location of each image region r_i in terms of its location in the panorama (top-left and bottom-right corners in normalized coordinates as well as area of the image covered), its elevation relative to the horizon, and its heading relative to the agents current and next viewing directions. All angles are encoded as $[\cos(\theta), \sin(\theta)]$. The resulting 11-dimensional vector S_i is projected into 2048 dimensions using a learned projection W^S. To capture the sequential order of the panoramas within a trajectory, we project the scalar panorama index to 2048 dimensions using a learned embedding W^P. As shown in Figure 4(b), the complete visual input representation for the image region is the element-wise sum of the visual features, panorama index embedding, and panoramic spatial embedding.

Extracting Image Regions from Panoramas. To extract image regions $\{v_1^{(i)}, \ldots, v_k^{(i)}\}$ from each panorama, we generate 600×600 pixel perspective projections using an 80 degree field of view at the 36 discrete heading and elevation directions used in previous work [4]. Similarly to ViLBERT, we use the bottom-up attention model [2,22] pretrained on Visual Genome [14] to extract a set of image regions and features from each perspective image. Since the perspective images have substantial overlap we remove redundant regions within each panorama. First, we discard regions that are centred more than 20 degrees away from the center of the image (i.e. we discard regions along the boarders). We assume that the discarded regions will be captured in a neighboring perspective image (spaced at 30 degree heading increments), with more visual context. Next,

we examine pairs of image regions within each panorama in order of decreasing feature similarity. We discard the region in the pair with the lower bottom-up attention class detection score, until a maximum of 100 regions per panorama remain. We define similarity as the cosine distance between image features to which we add the absolute difference in region heading and elevation. Including heading and elevation differences ensures that visually similar features found in different regions of the panorama are unlikely to be classified as redundant.

Training for Path Selection. To train VLN-BERT for path selection, we consider a 4-way multiple-choice task. Given an instruction x, we sample four trajectories out of which only one is successful $\{\tau_1^+, \tau_2^-, \tau_3^-, \tau_4^-\}$. We run VLN-BERT on each instruction-trajectory pair and extract their corresponding final representations. We denote these outputs for the CLS and the first IMG token as $h_{\texttt{CLS}}$ and $h_{\texttt{IMG}}$ respectively and compute a compatibility score s_i as

$$s_i = f(\tau_i, x) = W\left(h_{\texttt{CLS}}^{(i)} \odot h_{\texttt{IMG}}^{(i)}\right) \tag{5}$$

where $\odot$ denotes element-wise multiplication and W is a learned transformation matrix. Scores, normalized via a softmax, are supervised with cross-entropy loss,

$$\mathbf{p} = \text{softmax}(\mathbf{s}) \tag{6}$$

$$\mathcal{L}\left(x, \{\tau_1^+, \tau_2^-, \tau_3^-, \tau_4^-\}\right) = \text{CrossEntropy}\left(\mathbf{p}, \mathbf{1}[\tau_1^+]\right) \tag{7}$$

where $\mathbf{1}[\tau_1^+]$ is a 1-hot vector with mass at the index of τ^+. At inference, we simply sort trajectories by their compatibility scores s_i.

Mining Negative Examples. We find that choosing an appropriate set of path-instruction pairs is important for performance. Ideally, samples would span the space of all possible pairs, including hard negatives. The question is how to find varied path-instruction pairs with semantically meaningful differences? We find that using beam search with an instruction-following model yields a diverse set of paths that are effective for training. Specifically, we sample up to 30 beams per instruction from the follower model of Tan et al. [28] and label the path as successful if it meets the VLN success criteria (i.e. $< 3m$ from the goal). Finally, one positive and three negatives pairs are sampled at random for training.

4.3   Internet-to-Embodied Transfer Learning

While VLN-BERT can be trained from scratch, we designed the model to specifically enable transfer learning from language [7] and visiolinguistic [18] models trained on large-scale web corpora. This transfer is especially important due to the data-sparsity in VLN dataset (containing ∼14k path-instruction training pairs) and has a natural bias towards describing only objects present in training environments. In this section, we describe a pretraining curriculum for transferring models learned on 'disembodied' web data to the embodied VLN task.

We summarize the pretraining process in Figure 3. In total, we consider three stages focused on learning language, visual grounding, and action grounding.

- **Stage 1: Language.** We initialize the language stream of our model with weights from a BERT [7] model trained on Wikipedia and the BooksCorpus [34] under the masked language modelling and next sentence prediction objectives. Directly training on the path selection task after this stage is analogous to introducing a BERT encoder to represent instructions.
- **Stage 2: Visual Grounding.** Starting from a pretrained BERT model, Lu et al. train both streams of ViLBERT on the Conceptual Captions dataset [24] under the masked multimodal language modelling and multimodal alignment objectives. In this stage, we initialize model weights with a ViLBERT model trained in this manner. Training directly from this stage provides an initialization that can associate descriptions with image regions.
- **Stage 3: Action Grounding.** In the final stage, we pair paths and instructions from VLN and train the model under the masked multimodal modelling objective from [18]. While the previous stage learns to ground visual concepts, this stage additionally exposes the model to actions and their trajectory-based referents. For example, correctly predicting a masked instruction phrase like *'turn _____'* or *'stop at the _____'* requires the model to reason about the agent's path from the visual inputs and positional encodings.

Finally, we fine-tune VLN-BERT as described in the previous section.

5   Experiments

Our experiments primarily address following questions:

1. *Does pretraining on web image-text pairs improve VLN performance?*
2. *How does the performance of VLN-BERT compare with strong baselines?*
3. *Does VLN-BERT consider relevant image regions to produce alignment scores?*

5.1   Dataset

We conduct experiments using the Room-to-Room (R2R) navigation task [4] based on the Matterport3D dataset [5]. R2R contains human-annotated path-instruction pairs that are divided into training, seen and unseen validation, and unseen testing sets. To generate a dataset for path selection we run beam search on the instruction-follower model from [28], to produce a set of up to 30 candidate paths for each instruction in R2R. We find that with a beam size of 30 over 99% of the candidate sets contain one path that reaches the goal, which places an acceptable upper bound on path selection performance. In all of the experiments that follow, results are reported for selecting one path from the set of candidates.

5.2   Evaluation Metrics

We compare the performance of different models using standard VLN metrics.

– **Success rate (SR)** measures the percentage of selected paths that stop within 3m of the goal. In path selection this is our primary metric of interest.
– **Oracle Success rate (SR)** measures the percentage of selected paths with any position that passes within 3m of the goal.
– **Navigation error (NE)** measures the average distance of the shortest path from the last position in the selected path to the goal position.
– **Path length (PL)** measures the average length of the selected path.
– **Success rate weighted by path length (SPL)**, as defined in [1], provides a measure of success normalized by the ratio between the length of the shortest path and the selected path.

Note that for path selection we calculate metrics using only the selected path, which corresponds with the pre-explored environment setting. However, for the VLN leaderboard results we follow the required approach of prepending the exploration path to the selected path (which affects path length based metrics).

5.3   Training Baseline Models

We compare with the follower and speaker models from [28], which achieve state-of-the-art performance on the VLN test set in an ensemble model setting. The only auxiliary dataset used to train these baseline models is ImageNet [23] (used to pretrain an image feature extractor). All of the other components are trained from scratch (including word embeddings). Data augmentation, via environmental dropout [28], is used to train the follower model and greatly improves performance. We report results using code and weights provided by Tan et al. [28].

5.4   Results

Does pretraining on web image-text pairs improve VLN performance?
To answer this question we dissect our proposed training curriculum as indicated in Table 1, and find that in general each stage of training does contribute to performance. First, we find that our model has limited performance learning from scratch, achieving only 30.5% SR (compared with the 54.7% SR achieved by the speaker model from [28]). However, language-only pretraining, which corresponds to initializing our model with BERT [7] weights, improves performance substantially to 45.2% SR (an improvement of 14.7 absolute percentage points) – indicating that language understanding plays an important role in VLN.

Next, we find that both pretraining on image-text pairs from the Conceptual Captions [24] (visual grounding) and pretraining on path-instruction pairs from VLN [4] (action grounding) similarly improve success rate (by 4.5 and 4.9 absolute percentage points, respectively) when used independently. However, when the two pretraining stages are combined in series the improvement jumps to 14.1 absolute percentage points in success rate or 9.2 absolute percentage points over the next best setting. The substantial level of improvement that results from our full training curriculum suggests that not only does pretraining on webly-supervised image-text pairs from [24] improve path selection performance, but it also constructively supports the action grounding stage (Stage 3) of pretraining.

Table 1. We compare the contribution from the different stages of pretraining. We find that stage 2 and 3 both contribute significantly to task performance (improving Val Unseen Success Rate (SR) by ∼15-20 absolute percentage points over the no pretraining baseline – rows 3 and 4 vs. 1), with their full combination providing further synergistic gains (row 5). Notably, skipping the visual grounding stage (but still doing the others) results in a 9 absolute percentage point drop in Val Unseen SR (compare rows 4 and 5) – demonstrating the importance of internet-to-embodied transfer of visual grounding.

	#	Language Only	Visual Grounding	Action Grounding	PL	NE ↓	SPL ↑	OSR ↑	SR ↑	PL	NE ↓	SPL ↑	OSR ↑	SR ↑
		Pretraining Stage			**Val Seen**					**Val Unseen**				
VLN-BERT	1	(no pretraining)			10.78	6.78	0.35	54.22	37.55	10.29	6.81	0.27	50.62	30.52
	2	✓			10.33	4.89	0.55	69.31	58.73	9.59	5.47	0.41	57.34	45.17
	3	✓	✓		10.42	4.48	0.58	71.57	62.16	9.70	4.96	0.45	62.79	49.64
	4	✓		✓	10.51	4.28	0.60	72.65	63.82	9.81	5.05	0.46	62.75	50.02
	5	✓	✓	✓	10.28	3.73	0.66	76.47	70.20	9.60	**4.10**	**0.55**	69.22	**59.26**

How does VLN-BERT compare with strong baseline methods? The results in Table 2 compare path selection performance of VLN-BERT with the state-of-the-art speaker and follower models from [28]. We evaluate path selection using the set of up to 30 candidate paths generated with beam search using the follower from [28]. For the follower model results this amounts to taking the top beam from the candidate set. In the single model setting we see that VLN-BERT, trained with our full curriculum, achieves 59.3% SR, which is 4.6 absolute percentage points better than either of the other two methods.

In the pre-explored setting, the speaker and follower models are typically linearly combined (using a hyperparameter selected through grid search on val unseen) to further improve path selection performance [8, 28]. In the ensemble models section of Table 2, the speaker + follower line (row 4) represents our execution of the state-of-the-art ensemble model from [28]. In rows 5-7, we consider three model ensembles composed of a speaker, follower, and one additional model (again linearly combined with hyperparameters selected via grid search on val unseen). We find that adding another (randomly seeded) speaker or follower model yields modest improvements of 1.2 and 2.7 absolute percentage points in SR (rows 5 and 6). In contrast, adding VLN-BERT results in a 5.7 absolute percentage point boost in SR (row 7), which is 3.0 absolute percentage points higher on success rate than the next best ensemble.

In Table 3 we report results on the VLN test set via the VLN leaderboard, using the three-model ensemble that includes a speaker, follower, and VLN-BERT. The ensemble achieves a success rate of 73%, which is 4 absolute percentage points greater than previously published work [28], and 2 absolute percentage points greater than concurrent, unpublished work [33].

Table 2. Results comparing VLN-BERT with the follower and speaker from [28]. Notably, in the ensemble models setting, combining VLN-BERT with the speaker and follower results in a 3 absolute percentage point improvement in Val Unseen Success Rate (SR) over the next best three-model ensemble (compare rows 6 and 7).

	#	Re-ranking Model	Val Seen					Val Unseen				
			PL	NE ↓	SPL ↑	OSR ↑	SR ↑	PL	NE ↓	SPL ↑	OSR ↑	SR ↑
Single Models	1	follower (flw) [28]	10.40	3.68	0.62	74.12	65.10	9.57	5.20	0.49	58.79	52.36
	2	speaker (spk) [28]	11.19	3.80	0.56	77.25	60.69	10.71	4.25	0.49	**72.07**	54.66
	3	VLN-BERT	10.28	3.73	0.66	76.47	70.20	9.60	**4.10**	**0.55**	69.22	**59.26**
Ensemble Models	4	spk + flw [28]	10.69	2.72	0.70	82.94	74.22	10.10	3.32	0.63	76.63	67.90
	5	spk + flw + flw	10.73	2.72	0.71	83.33	74.71	10.12	3.22	0.64	77.56	69.14
	6	spk + flw + spk	10.77	2.45	0.73	85.98	76.86	10.17	2.99	0.65	79.28	70.58
	7	spk + flw + VLN-BERT	10.61	2.35	0.78	86.57	81.86	10.00	**2.76**	**0.68**	**81.91**	**73.61**

Table 3. Leaderboard results on Test Unseen for methods using beam search.

Re-ranking Model	Test Unseen				
	PL	NE ↓	SPL ↑	OSR ↑	SR ↑
Speaker-Follower [8]	1,257	4.87	0.01	96	53
Tactical Rewind [13]	197	4.29	**0.03**	90	61
Self-Monitoring [20]	373	4.48	0.02	97	61
Reinforced Cross-Modal Matching [30]	358	4.03	0.02	96	63
Environmental Dropout [28]	687	3.26	0.01	**99**	69
Auxiliary Tasks† [33]	41	3.24	0.21	81	71
VLN-BERT	687	**3.09**	0.01	**99**	**73**

†indicates unpublished/concurrent work

Does VLN-BERT consider relevant image regions to produce alignment scores? To gain insight into the visual grounding learned by VLN-BERT, we visualize which panoramic image regions affect the compatibility score. This analysis is performed using a simple gradient-based visualization technique [25]. We take the gradient of our learned score $f(x, \tau)$ with respect to the feature representation for each region from each panorama, and sum this 2048-dimensional gradient vector over the feature dimension to produce a scalar measure of region importance. To gain further insight, we analyze how the region importance varies when the instruction is perturbed by removing parts of the description.

Two examples of this analysis are illustrated in Figure 5. The left panel provides the original and modified versions of the instructions. The middle panel illustrates the region importance histograms and the top-5 regions that influence the compatibility score for the original instructions. The right panel provides the equivalent illustrations for the modified instructions. The words and regions highlighted in green provide a qualitative assessment of the visual grounding learned by VLN-BERT. Next, we remove a part of the original instruction that refers to a high-importance region, producing the modified versions of the instructions. The purple highlights demonstrate that after the instructions have
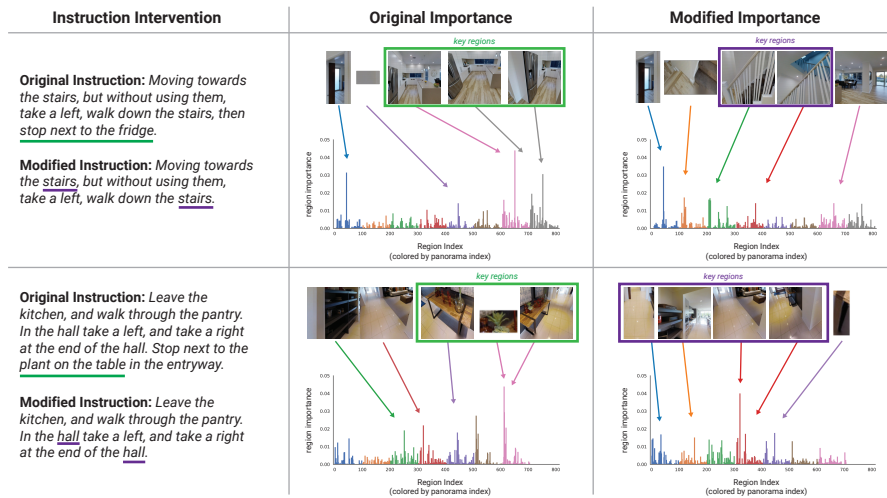
Fig. 5. We compare region importance histograms under instruction ablations – removing a phrase or sentence. Region importance is calculated by taking the gradient of the compatibility score with respect to image region features. The images above each histogram correspond to the most influential regions (i.e. the peaks in the histogram). The underlined instruction phrases correspond with the regions outlined in green and purple. In the top example, removing the reference to the *'fridge'* (in green) shifts the importance to other regions along the path (i.e. the *'stairs'* in purple), suggesting that VLN-BERT considers visually relevant image regions to score path-instruction pairs.

been modified, VLN-BERT appropriately shifts importance to remaining scene elements referenced in the instructions. For example, in the first row regions containing a *'fridge'* are important for the original instruction, whereas for the modified instruction the importance shifts to the *'stairs'*.

6 Conclusion

In this work, we demonstrated internet-to-embodied transfer of visual concept grounding – leveraging large-scale image-text data from the web to improve a discriminative path-instruction alignment model for VLN. In our path re-ranking setting, this model improves over prior work and our ablations show each stage of our transfer curriculum contributes significantly.

Acknowledgements

References

1. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757 (2018)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018)
3. Anderson, P., Shrivastava, A., Parikh, D., Batra, D., Lee, S.: Chasing ghosts: Instruction following as bayesian state tracking. In: Advances in Neural Information Processing Systems. pp. 369–379 (2019)
4. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
5. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. International Conference on 3D Vision (3DV) (2017)
6. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: Advances in Neural Information Processing Systems. pp. 3314–3325 (2018)
9. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
10. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. arXiv preprint arXiv:2002.10638 (2020)
11. Huang, H., Jain, V., Mehta, H., Baldridge, J., Ie, E.: Multi-modal discriminative model for vision-and-language navigation. arXiv preprint arXiv:1905.13358 (2019)
12. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.L.: Referit game: Referring to objects in photographs of natural scenes. In: EMNLP (2014)
13. Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., Srinivasa, S.: Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6741–6749 (2019)
14. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. arXiv preprint arXiv:1602.07332 (2016)
15. Li, G., Duan, N., Fang, Y., Jiang, D., Zhou, M.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. arXiv preprint arXiv:1908.06066 (2019)

16. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 (2019)
17. Li, X., Li, C., Xia, Q., Bisk, Y., Celikyilmaz, A., Gao, J., Smith, N., Choi, Y.: Robust navigation with language pretraining and stochastic sampling. arXiv preprint arXiv:1909.02244 (2019)
18. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Advances in Neural Information Processing Systems. pp. 13–23 (2019)
19. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances in neural information processing systems. pp. 289–297 (2016)
20. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. arXiv preprint arXiv:1901.03035 (2019)
21. Ma, C.Y., Wu, Z., AlRegib, G., Xiong, C., Kira, Z.: The regretful agent: Heuristic-aided navigation through progress estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6732–6740 (2019)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
24. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
25. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
26. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: Vl-bert: Pre-training of generic visual-linguistic representations. arXiv preprint arXiv:1908.08530 (2019)
27. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP (2019)
28. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. arXiv preprint arXiv:1904.04195 (2019)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
30. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6629–6638 (2019)
31. Wang, X., Xiong, W., Wang, H., Yang Wang, W.: Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 37–53 (2018)
32. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J.J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. arXiv preprint arXiv:1909.11059 (2019)

33. Zhu, F., Zhu, Y., Chang, X., Liang, X.: Vision-language navigation with self-supervised auxiliary reasoning tasks. arXiv preprint arXiv:1911.07883 (2019)
34. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision. pp. 19–27 (2015)