A New Path: Scaling Vision-and-Language Navigation with Synthetic Instructions and Imitation Learning

Anonymous authors
Paper under double-blind review

Abstract

Recent studies in Vision-and-Language Navigation (VLN) train RL agents to execute natural-language navigation instructions in photorealistic environments, as a step towards intelligent agents or robots that can follow human instructions. However, given the scarcity of human instruction data and limited diversity in the training environments, these agents still struggle with complex language grounding and spatial language understanding. Pretraining on large text and image-text datasets from the web has been extensively explored but the improvements are limited. To address the scarcity of in-domain instruction data, we investigate large-scale augmentation with synthetic instructions. We take 500+ indoor environments captured in densely-sampled 360° panoramas, construct navigation trajectories through these panoramas, and generate a visually-grounded instruction for each trajectory using Marky (Wang et al., 2022), a high-quality multilingual navigation instruction generator. To further increase the variability of the trajectories, we also synthesize image observations from novel viewpoints using an image-to-image GAN. The resulting dataset of 4.2M instruction-trajectory pairs is two orders of magnitude larger than existing human-annotated datasets, and contains a wider variety of environments and viewpoints. To efficiently leverage data at this scale, we train a transformer agent with imitation learning for over 700M steps of experience. On the challenging Room-across-Room dataset, our approach outperforms all existing RL agents, improving the state-of-the-art NDTW from 71.1 to 79.1 in seen environments, and from 64.6 to 66.8 in unseen test environments. Self-supervision with synthetic instructions in new environments can improve further to 68.6 (vs. human 79.5). Our work points to a new path to improving instruction-following agents, emphasizing large-scale imitation learning and the development of synthetic instruction generation capabilities – which are shown to flow through directly to improved instruction-following performance.

1 Introduction

Developing intelligent agents that follow human instructions is a long-term, formidable challenge in AI (Winograd, 1971). A recent focus addressing this problem space is Vision-and-Language Navigation (VLN) (Anderson et al., 2018b; Chen et al., 2019). Navigation is an ideal test bed for studying instruction-following, since the task can be simulated photo-realistically at scale and evaluation is straightforward. However, as in all instruction-following tasks, datasets that capture the linguistic diversity and idiosyncrasies of real human instructors are small and expensive to collect.

Shortages of human-annotated training data for other vision-and-language tasks have been partially addressed by pretraining transformers on up to billions of image-text pairs. This has underpinned dramatic improvements in image captioning (Zhang et al., 2021; Wang et al., 2021), visual question answering (Tan & Bansal, 2019), phrase grounding (Kamath et al., 2021; Li et al., 2022c), text-to-video retrieval, video question answering (Lei et al., 2021) and text-to-image synthesis (Ramesh et al., 2021; Yu et al., 2022). However, these are all static image or video tasks, whereas VLN agents *interact* with 3D environments. In VLN, pretraining on large image-text and text-only datasets has been thoroughly explored (Majumdar et al., 2020; Hong et al., 2020; 2021; Moudgil et al., 2021), but improvements are more limited. Zhu et al. (2022) argue that progress in VLN has plateaued while still leaving a large gap between machine and human performance. We hypothesize that static image-text and text-only datasets – despite their size – lack the spatially grounded and action-oriented language needed for effective VLN pretraining. Consider instructions from the Room-across-Room (RxR) dataset (Ku et al., 2020), which illustrate that successful wayfinding requires an understanding of allocentric and egocentric spatial expressions (*near a grey console table **behind you***), verbs (***climb** the stairs*), imperatives and negations (***do not** enter the room in front*) and temporal conditions (*walk **until** you see an entrance on your left*). Such expressions are rarely found in image-text datasets. Though similar expressions are found in text-only corpora, their meaning as it relates to the physical world is hard to infer from text alone (without sensorimotor context) (Bisk et al., 2020).

To address this problem, we investigate large-scale augmentation with synthetic in-domain data, i.e., model-generated navigation instructions for trajectories in realistic 3D environments. For this, we construct a large dataset using Marky (Wang et al., 2022), which generates VLN instructions that approach the quality of human instructors. Wang et al. (2022) released the 1M Marky instruction-trajectory pairs situated in 61 Matterport3D (Chang et al., 2017) environments. To increase the diversity of the environments (and thus the scenes and objects available in them), we automatically annotate an additional 491 environments from the Gibson dataset (Xia et al., 2018). Gibson environments have been underutilized in prior VLN work due to the lack of navigation graphs indicating navigable trajectories through its densely-sampled 360° panoramas. We train a model that classifies navigable directions for Matterport3D and use it to construct the missing navigation graphs. We sample 3.2M trajectories from these graphs and annotate them with Marky. To further increase the variability of trajectories, we synthesize image observations from novel viewpoints using an image-to-image GAN (Koh et al., 2022). The resulting dataset is two orders of magnitude larger than existing human-annotated ones, and contains a wider variety of scenes and viewpoints.

With orders of magnitude more training examples and environments, we explore VLN agent performance with imitation learning (IL), i.e., behavioral cloning and DAGGER (Ross et al., 2011). IL can take advantage of high-throughput transformer frameworks such as T5 (Raffel et al., 2020) and thus efficiently train on 4.2M instructions (accumulating over 700M steps of experience). This is a major departure from prior VLN work in low-data settings, e.g. Chen et al. (2021) report that pure IL underperforms by 8.5% success rate compared to agents trained with both IL and online reinforcement learning (RL) algorithms such as A3C (Mnih et al., 2016). However, IL outperforms RL in related tasks with sufficient training data (Ramrakhya et al., 2022). Online RL also requires interacting with the environment at each step; this precludes efficient data prefetching and parallel preprocessing and thus imposes unavoidable overhead compared to IL. Empirically, we confirm that training existing models such as HAMT (Chen et al., 2021) on 4.2M instructions is infeasible without ground-up re-engineering, though we do find incorporating 10K additional synthetic instructions into HAMT training modestly improves performance. Training with IL aligns with the trend towards large-scale multi-task vision-and-language models trained with supervised learning; these have unified tasks as diverse as visual question answering, image captioning, object detection, image classification, OCR and text reasoning (Chen et al., 2022) – and could include VLN in future.

Experimentally, in detailed ablation studies we show that adding Gibson environments, synthesizing additional image observations from novel viewpoints, increasing the capacity of the transformer, and finetuning with DAGGER all improve agent performance. On the challenging RxR dataset – which contains multilingual instructions with a median trajectory length of 15m – our best agent *using only imitation learning* outperforms all prior RL agents. Evaluating on novel instruction-trajectories in seen environments (Val-Seen), we improve over the state-of-the-art by 8%, reaching 79.1 NDTW. In new, unseen environments (Test), we improve by 2%, achieving 66.8 NDTW. We also show that that self-training with synthetic instructions in new environments (still without human annotations) improves performance by an additional 2% to 68.6 NDTW. Overall, our RxR results point to a new path to improving instruction-following agents, emphasizing large-scale imitation learning and the development of synthetic instruction generation capabilities. Perhaps surprisingly, on the English-only R2R dataset (Anderson et al., 2018b), our imitation-learning agent achieves strong performance but not state-of-the-art. Marky was trained on RxR, so we attribute this to domain differences between R2R and RxR, underscoring the domain dependence of synthetic instructions.

In summary, the contributions of this work are threefold:

- Navigation graphs for Gibson environments, used to generate a new large-scale synthetic dataset of 3.2M multilingual navigation instructions (which we plan to release).

- Testing the limits of pure imitation learning for training VLN agents, including detailed ablation studies exploring massive synthetic data augmentation.
- Substantial gains in the state-of-the-art on the challenging RxR dataset, demonstrating that advances in instruction generation flow directly into instruction-following performance.

2 Related Work

Vision-and-Language Navigation Agents that attempt to follow instructions by navigating to a prescribed location were initially studied in simple environments requiring limited or no perception, using instructions that were often procedurally generated (Chen & Mooney, 2011; Artzi & Zettlemoyer, 2013; Andreas & Klein, 2015; Mei et al., 2016; Misra et al., 2017). More recent work has included settings based on photorealistic 3D environments and natural language instructions (Anderson et al., 2018b; Chen et al., 2019), using environments such as Matterport3D (Chang et al., 2017) and Streetview (Mehta et al., 2020). This instantiation of the problem, known as Vision-and-Language Navigation (VLN), raises the prospect of sim-to-real transfer to physical robots (Anderson et al., 2021), and encouraged further datasets exploring dialog (Thomason et al., 2019; Hahn et al., 2020), search for objects (Qi et al., 2020) and multilingual instructions (Ku et al., 2020).

Pretraining and Transfer Learning The use of realistic imagery and language in VLN, combined with the cost of collecting human instruction annotations, leads to a natural focus on pretraining and transfer learning to improve performance. Majumdar et al. (2020) formulate VLN as an instruction-trajectory alignment problem, and initialize a transformer model using pretrained BERT weights (Devlin et al., 2019) then perform additional pretraining on image-text pairs from Conceptual Captions (Sharma et al., 2018). Li et al. (2019b) also use a BERT model, although more recent approaches have favored learned text encoders from scratch by pretraining with Masked Language Modeling (MLM) and related objectives on instruction-trajectory data (Hao et al., 2020; Chen et al., 2021). In terms of image representations, early work (Fried et al., 2018) used ResNet features (He et al., 2016) pretrained on ImageNet (Russakovsky et al., 2015), although pretrained object detectors have also been explored (Hong et al., 2021; Majumdar et al., 2020; Li et al., 2022b) (typically a Faster-RCNN (Ren et al., 2015)). More recently, Chen et al. (2021) use a vision transformer (ViT) (Dosovitskiy et al., 2021) and current-state-of-the art agents (Shen et al., 2021; Li et al., 2022a) use CLIP (Radford et al., 2021), obtaining improvements over similarly sized encoders pretrained on ImageNet. However, although pretraining and transfer learning from large text and image-text datasets has been thoroughly explored, a significant gap to human performance remains.

Data Augmentation Fried et al. (2018) were the first to show that performance following human instructions could be improved by augmenting training with synthetic (model-generated) instructions. However, in human wayfinding evaluations, the instructions used (Fried et al., 2018; Tan et al., 2019) were shown to be surprisingly weak, being poorly grounded and mostly unfollowable by people (Zhao et al., 2021). To reduce hallucinations and improve instruction quality, Wang et al. (2022) proposed Marky, a stronger instruction generator trained with text-aligned visual landmark correspondences. We are the first to investigate data augmentation with Marky, which we compare to Fried et al. (2018) (Marky is superior). On the other hand, Tan et al. (2019) and Li et al. (2022a) perform data augmentation by modifying existing environments before generating new instructions (an approach which may be complimentary to ours). Guhur et al. (2021) train on a synthetic dataset of path-instruction pairs generated using online rental listings. Koh et al. (2022) train a generative model that infills and outpaints spatially perturbed panos of indoor environments (see Section 5).

3 Approach

Problem set up The agent is instantiated in an environment and must follow a natural language instruction $\mathcal{W}$. At time step t, the agent receives observation o_t and chooses action a_t that transitions it from state s_t to new state s_{t+1}. Following prior work, each observation is a photorealistic panoramic image (hereafter, pano) encoded as 36 image feature vectors $o_t{=}\{I^o_{t,1}, I^o_{t,2}, ..., I^o_{t,K}\}$. These features are extracted from perspective projections at 36 view angles (12 headings × 3 elevations at 30° intervals). The agent moves by choosing an action a_t from a set of candidates $\mathcal{A}_t{=}\{I^a_{t,1}, I^a_{t,2}, ..., I^a_{t,J}\}$ given by the environment. Action candidates are determined by the adjacent panos in a predefined navigation graph; each is represented by the image feature vector extracted from the perspective pro-

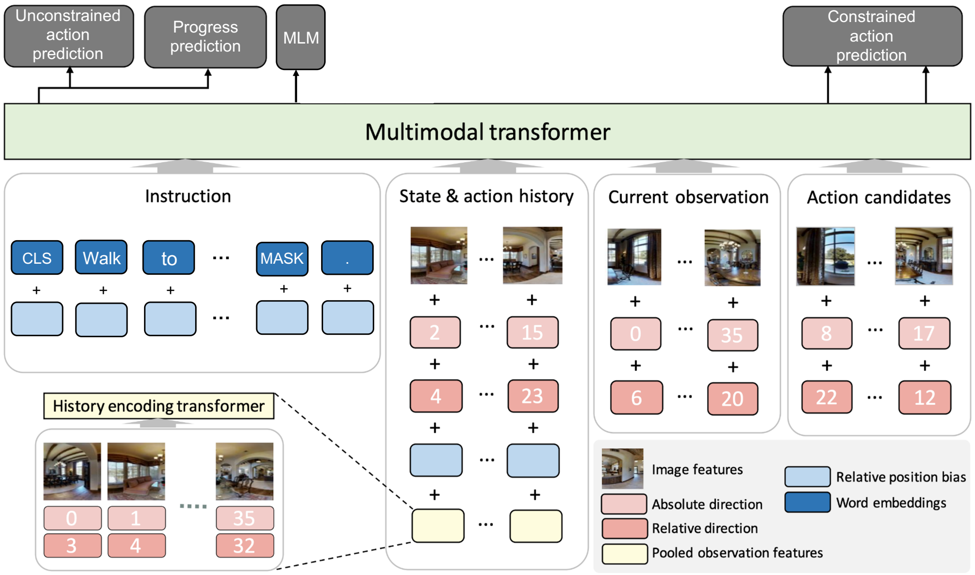


Figure 1: Agent architecture. At each time step, we combine the instruction, state and action history, current observation and action candidates into a multimodal transformer to predict the next action. Since observations consist of 36 image feature vectors (representing different views from a 360° camera) we compress each previous observation into a single vector, similar to Chen et al. (2021).

jection looking towards the adjacent pano. Selecting an action teleports the agent a few meters to the new pano. Alternatively, the agent can choose 'STOP' to end the episode. On average agents have 5 actions available at each step, including 'STOP'. See Anderson et al. (2018b) for more details.

Agent architecture Our imitation-learning agent is a transformer encoder which predicts the next action a_{t+1} by jointly combining all four input modalities: the instruction text $\mathcal{W}$, the history of observations $o_{1:t-1}$ and actions $a_{1:t-1}$, the current observation o_t, and the action candidates $\mathcal{A}_t$ (see Figure 1). At each step, all input features are concatenated into a single multimodal sequence with no attention masking, allowing every input to attend to every other input. For biasing interactions between different input modalities we include learned attention biases for each pair of input types, e.g. the instruction and the observation/action history. Like HAMT (Chen et al., 2021), our approach is not autoregressive: every forward pass predicts a single action using the full history. Given our emphasis on data augmentation, we name our agent MARVAL for Maximum Augmentation Regime for Vision And Language navigation. Our implementation is based on mT5 (Xue et al., 2021), a multilingual variant of the T5 transformer architecture (Raffel et al., 2020).

Image features As noted above, pano observations o_t and action candidates $\mathcal{A}_t$ are represented with sets of image features. We use precomputed, fixed 640-d features from MURAL-large (Jain et al., 2021), an EfficientNet-B7 (Tan & Le, 2019) backbone trained on 1.8B multilingual image-text pairs and 6B translation pairs. MURAL's image encoder's representational power is similar to CLIP's (Radford et al., 2021), which is used in previous work (Shen et al., 2021; Chen et al., 2021) and is trained on 400M English image-text pairs with a VIT (Dosovitskiy et al., 2021) backbone. To provide orientation information, each feature is combined with two learned embeddings: an *absolute direction embedding* capturing the feature's orientation in the environment's fixed coordinate system, and a *relative direction embedding* based on orientation relative to the agent's heading. Note that the agent's initial heading at $t{=}0$ is given by the dataset, and is typically random. We also augment the set of action candidates $\mathcal{A}_t$ with a 'STOP' action. This is convenient for modeling action classification over the candidates (refer 'Action classification', below) and is represented by a zero image vector with unique direction embeddings. We use 37 absolute and relative direction embeddings, and snap features to the closest.

Instruction encoding The instruction $\mathcal{W}$ is encoded as a sequence of WordPiece (Sennrich et al., 2016) tokens using the mT5 vocabulary which supports up to 101 languages via a Sentence-

Piece (Kudo & Richardson, 2018) model trained on mC4. Following T5, position information within the instruction is derived from relative position biases applied to the transformer's attention logits.

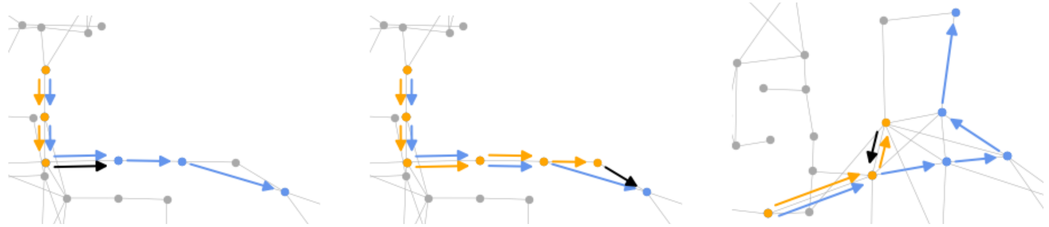
History encoding The history of agent observations $o_{1:t-1}$ and actions $a_{1:t-1}$ is computationally expensive to process, since each pano observation o_t is comprised of 36 feature vectors. Similar to Chen et al. (2021) we embed the 36 features from each previous pano observation into a single vector, based on the mean-pooled output of a separate transformer applied to the image features and their direction embeddings. This is added to the action candidate selected at each previous step. Position information for the state and action history is provided by relative position biases.

Pretraining We train the agent in two stages. We first pretrain on a large dataset of instruction-trajectory pairs, including both model-generated instructions and trajectories containing synthesized image observations from novel viewpoints (refer Section 4). We then finetune on a single dataset of human-annotated instruction-trajectory pairs to maximize performance on that dataset. Unlike Majumdar et al. (2020) and Li et al. (2019a), our transformer weights are initialized from scratch – we do not use any image-caption datasets or text corpora to train the transformer. Since the model is not autoregressive, each training trajectory is broken down into T training examples, where T is the number of time steps in the trajectory. Each training example requires the model to predict the next action for a single step in a trajectory, given the full instruction $\mathcal{W}$, the action history $a_{1:t-1}$, the observation history $o_{1:t-1}$, the current observation o_t and the set of action candidates $\mathcal{A}_t$. To increase the amount of supervision, during pretraining we combine four tasks:

- **Masked language modeling (MLM)** (Devlin et al., 2019): 15% of tokens in the instruction are masked, with all consecutive spans of masked tokens replaced by a single MASK token. Similar to Chen et al. (2021), the model predicts the masked tokens using the surrounding text and visual clues from the observation/action history and the current observation.
- **Progress prediction**: A small MLP is added to the output representation of the CLS token (a special symbol capturing the fused representation of the entire sequence) to predict the proportion of the trajectory that is completed (based on 20 discretized classes). Progress monitoring has been shown to improve instruction grounding (Ma et al., 2019).
- **Constrained action prediction**: A classification task to predict the correct action from the constrained set of available action candidates $\mathcal{A}_t$. Since action candidates are inputs to the encoder (refer Figure 1), we compute the logit for each action as a learned projection of its output representation and normalize with softmax (a simplification of Chen et al. (2021)).
- **Unconstrained action prediction**: A second small MLP is added to the CLS output to directly predict the next action from all 36 discretized agent-relative directions or 'STOP'. Hence, these predictions are not constrained to $\mathcal{A}_t$, similar to the approach in Hao et al. (2020). The constrained and unconstrained action prediction tasks are highly related but complimentary; in early experiments we found that equally weighting the logits from both improves accuracy by 1-2%, so we adopt this approach in all experiments.

Finetuning (behavioral cloning) During finetuning we adapt our pretrained agent for best performance on a smaller human-annotated dataset. We update only the WordPiece embeddings in the agent and keep all other transformer weights frozen, as this makes finetuning more stable and less prone to overfitting (especially on the smaller R2R dataset). We consider two finetuning strategies. The first is *behavioral cloning*. In this setting, we simply drop the instruction text masking and the MLM objective, retaining the progress prediction and constrained and unconstrained action prediction losses used in pretraining. We then finetune the agent to predict the next action at each step along ground-truth trajectories, treating imitation learning as supervised learning.

DAGGER training The main weakness of behavioral cloning is that the state distribution seen in training differs from the state distribution induced by the agent during inference (Ross & Bagnell, 2010). Previous works (Chen et al., 2021; Tan et al., 2019) report substantial improvements by combining behavioral cloning with online reinforcement learning algorithms such as A3C (Mnih et al., 2016). We use *DAGGER* (Ross & Bagnell, 2010) to help train the agent to better recover from errors, since it is simple to implement and requires no environment interaction during training. In DAGGER, during each iteration of finetuning the dataset is augmented with trajectories of states visited by the current agent policy and actions given by an expert. Figure 2 explains the calculation of expert actions. We find that most of the gains are captured in a single DAGGER iteration.



(a) Agent is on the GT trajectory: Expert selects the next action in the GT trajectory.

(b) Agent is off the GT trajectory; GT trajectory is the shortest-path from start to goal: Expert action is the first step in the recalculated shortest-path to the goal.

(c) Agent is off the GT trajectory; GT trajectory is not a shortest-path: Expert takes the shortest path back to the closest node in the GT trajectory.

Figure 2: DAGGER training: Calculation of the expert action (in black) given the ground-truth (GT) trajectory (in blue) and an agent trajectory (in yellow).

Pre-Exploration While most of the focus in VLN is on instruction-following in new, unseen environments, in reality environments persist over time providing opportunities for pre-exploration. Similar to Tan et al. (2019); Wang et al. (2019) we consider a *pre-exploration* setting in which the agent may explore unseen environments with self-supervision before evaluation. Our synthetic-instruction approach is readily applicable to this scenario; we simply sample paths from the Val-Unseen or Test environments, annotate them with Marky instructions, and include them in the training data.

4 Datasets and Augmentation

The datasets used for training and evaluation are described below and summarized in Table 1.

- **Room-to-Room (R2R)** (Anderson et al., 2018b) consists of 22K human-annotated English language navigation instructions, each describing a trajectory that traverses multiple rooms in Matterport3D (Chang et al., 2017). This was the first dataset to use a photo-realistic environment for the instruction guided navigation task. R2R trajectories average 10m, and the trajectories are always the shortest path between the start point and the goal.
- **Room-across-Room (RxR)** Ku et al. (2020) introduce RxR, a larger human-annotated dataset containing 126K instructions in English, Hindi and Telugu. To mitigate goal seeking behaviour and to ensure that agents are faithful to the instruction, RxR includes Matterport3D trajectories that are diverse in terms of length (average is 15m) and the landmarks that are referred to, and it also includes trajectories that do not go directly to the goal.
- **Speaker-Matterport (S-MP)** Fried et al. (2018) train an LSTM (Hochreiter & Schmidhuber, 1997) Speaker model on ground-truth navigation instructions from R2R, and use this model to generate synthetic instructions for a set of newly sampled trajectories in Matterport3D environments, generating 178k new instruction-path pairs.
- **Marky-Matterport (M-MP)** Marky is a landmark-aware multilingual instruction generator trained on RxR. Wang et al. (2022) use it to generate 1M instructions in English, Hindi and Telugu for 330K sampled Matterport3D trajectories. In human wayfinding evaluations in unseen environments Marky achieves close to human performance on shortest-path trajectories (e.g., R2R's paths). On the more challenging RxR paths a gap remains: human wayfinders obtain a 62% success rate with Marky vs. 78% with human instructions.
- **Marky-Gibson (M-Gib)** The Gibson (Xia et al., 2018) dataset consists of 572 indoor 3D environments. Despite its large size compared to Matterport3D, prior work has underutilized Gibson data for training VLN agents. This is primarily due to a lack of navigation trajectories and instruction annotations. To alleviate this, and unlock the Gibson dataset for VLN training, we propose an automated process to label these environments with high quality navigation graphs, described in detail below. We then sample 1.3M trajectories and annotate them with 3.2M Marky instructions in English, Hindi and Telugu.

Gibson navigation graphs In the standard VLN setting, agents are trained and evaluated using panos as observations (refer Section 3). Movement in the environment requires a graph with panos as nodes and edges indicating navigability. Navigation graphs for Matterport3D were generated

Dataset	Instruction Count	Avg steps	Avg words	Environment	Model-generated	Lang
R2R (Anderson et al., 2018b)	14K	5.0	26	Matterport	✗	en
RxR (Ku et al., 2020)	79K	8.1	78	Matterport	✗	en/hi/te
Speaker-Matterport (Fried et al., 2018)	178K	5.1	21	Matterport	✓	en
Marky-Matterport (Wang et al., 2022)	1.0M	9.5	87	Matterport	✓	en/hi/te
Marky-Gibson (ours)	3.2M	7.1	71	Gibson	✓	en/hi/te

Table 1: Training data. Existing datasets are situated in the 61 train environments from Matterport3D (Chang et al., 2017). We introduce the multilingual Marky-Gibson dataset containing 3.2M model-generated navigation instructions in 491 Gibson (Xia et al., 2018) environments.

by Anderson et al. (2018b), using a semi-automated process combined with human visual inspection. However, there are no navigation graphs for Gibson environments and the size of the dataset precludes human inspection. We therefore train a model on panos and navigation graphs from the Matterport3D train split to classify whether a patch of pixels in a pano constitutes a navigable direction. The model is based on RedNet (Jiang et al., 2018), an RGB-D encoder-decoder first proposed for image segmentation, using a ResNet-50 (He et al., 2016) backbone. The output space is discretized into $8 \times 16 \times 5$ pitch, heading and distance buckets. During training each bucket is assigned a positive value if the corresponding location corresponds to a navigable node, and 0 otherwise.

To compute Gibson navigation graphs, we combine model edge predictions with obstacle information from the dataset's 3D meshes. We add an edge between pano nodes i and j if the following boolean expression evaluates to true:

$$e(i,j) = (\lambda_d \frac{g_{i,j}}{s_{i,j}} - \lambda_p p_{i,j} \leq 1) \wedge (s_{i,j} \leq 3.5) \wedge (|z_i - z_j| \leq 3) \tag{1}$$

where $g_{i,j}$ is the geodesic distance (accounding for obstacles) between nodes i and j calculated using the Habitat Simulator (Savva et al., 2019), $s_{i,j}$ is the straight-line Euclidean distance between nodes i and j, $p_{i,j}$ is the model probability of an edge connecting nodes i and j, z_i is the vertical coordinate of pano i, and λ_d and λ_p are weighting parameters. The first term captures model predictions and encourages edges between panos that have few intervening obstacles. The second term ensures that nodes are within 3.5m, and the third term ensures that nodes are within 3m in the vertical axis. Finally, to ensure that the navigation graph for each environment is fully connected, we compute the minimum spanning tree (MST) (Kruskal, 1956) of the graph with the edge weights given by the first term in Equation 1, and apply a logical 'OR' operation over $e(i,j)$ and the MST. To set the weighting parameters λ_d and λ_p, we perform grid search to maximize the F_1 score when predicting edges in Matterport3D val environments. Our approach achieves an F_1 score of 0.70, precision of 0.695, and recall of 0.713. The average edge length in the generated Gibson graphs is 3.02m (median of 2.06m), and the average node degree is 4.15 (median of 4).

Trajectory sampling and instruction generation Using the generated navigation graphs, we sample trajectories from 491 Gibson train and val environments (we do not use test environments). Unlike Matterport3D, Gibson lacks room annotations, which precludes us from using the two-step sampling approach from RxR. Instead, we use a simpler approach: we randomly sample 3 panos, and use a TSP solver to find the shortest path that visits all 3 panos. Trajectories longer than 40m or 16 steps are discarded, and no more than 3K paths are sampled per environment. This procedure generates 1.06M paths, with an average of 9.5 steps and length of 28.6m. Using Marky we annotate each trajectory with English, Hindi and Telugu instructions to create the Marky-Gibson dataset.

Synthesizing image observations with SE3DS One weakness of training VLN agents on pano images is that training trajectories are constrained to the locations of the captured images. VLN agents tend to overfit to these trajectories (Zhang et al., 2020), contributing to a performance drop in unseen environments. Koh et al. (2021; 2022) showed that a strong generative model is capable of successfully rendering high resolution panos from novel viewpoints, and that training VLN agents with spatially-perturbed panos could improve the success rate of the agent on R2R Val-Unseen by 1.5%. To assess if this approach is complimentary to instruction augmentation, we use the proposed SE3DS (Simple and Effective 3D Synthesis) model to augment panoramas from the Matterport environments. Following Koh et al. (2022), we create 200 variations of each environment which are randomly sampled during training. In each environment variation, with 50% probability a pano will be spatially-perturbed by up to 1.5m and re-rendered at the new location using SE3DS.

		Pretraining Data							RxR Val-Unseen				R2R Val-Unseen		
	Size	R2R	RxR	S-MP	M-MP	M-Gib	SE3DS	Its	NE	SR	NDTW	SDTW	NE	SR	SPL
1	Base	✓	✓					630K	11.16	23.2	40.1	19.0	7.58	33.5	31.9
2	Base	✓	✓	✓				1.68M	11.17	25.0	40.3	21.0	6.50	41.0	39.0
3	Base	✓	✓	✓	✓			2.94M	7.53	45.0	56.9	39.2	7.07	40.3	37.2
4	Base	✓	✓		✓			2.00M	6.90	50.2	60.3	43.9	6.07	50.0	46.5
5	Base	✓	✓	✓	✓		✓	1.94M	6.68	50.9	61.6	45.0	5.34	52.2	49.0
6	Base	✓	✓	✓	✓	✓		3.00M	5.90	55.6	64.4	49.0	5.24	52.5	47.9
7	Base	✓	✓	✓	✓	✓	✓	2.80M	5.75	55.9	65.1	49.3	5.08	54.2	50.1
8	Large	✓	✓	✓	✓	✓	✓	4.80M	5.72	58.3	66.1	51.7	4.87	55.6	52.8
9	Large	✓	✓		✓	✓	✓	5.14M	**5.56**	**59.4**	**67.0**	**52.7**	**4.84**	**57.4**	**54.6**

Table 2: Comparison of pretraining settings. Best results (without finetuning) are obtained by combining the **R2R** and **RxR** datasets *with* Marky-generated instructions in both Matterport3D (**M-MP**) and Gibson (**M-Gib**) environments, *without* Speaker-generated instructions (**S-MP**), and *with* synthesis of observations from novel viewpoints using **SE3DS** (row 9).

5 Experiments

Pretraining settings In Table 2 we explore pretraining using varying amounts and types of augmented data. During pretraining, we monitor one-step action prediction accuracy on ground-truth trajectories using held-out instructions from RxR and R2R Val-Unseen. Each setting is trained until convergence, requiring more iterations (**Its**) for larger models and datasets. We select the best checkpoint based on one-step prediction accuracy then perform a full evaluation using standard VLN path-fidelity metrics (Anderson et al., 2018a; Ilharco et al., 2019): Navigation Error (**NE** ↓, the average distance in meters between the agent's final position and the goal), Success Rate (**SR**↑, the proportion of trajectories with NE $<$ 3m), Success rate weighted by normalized inverse Path Length (**SPL** ↑), normalized dynamic time warping (**NDTW**↑), and success weighted DTW (**SDTW**↑).

Speaker vs. Marky Consistent with previous work, we find that data augmentation with synthetic instructions from the Fried et al. (2018) Speaker model improves performance (row 2 vs. 1, +2% SR on RxR and +8% on R2R), but instructions from Marky (Wang et al., 2022) are far more effective (row 4 vs. 1, +27% SR on RxR and +17% on R2R). This is consistent with human evaluations of instruction quality, confirming that improvements in instruction-generation flow through to instruction-following performance. Interestingly, we find that combining the Speaker model with Marky leads to worse performance on both RxR and R2R (row 3 vs. 4, and also row 8 vs. 9), which we attribute to the introduction of noise from the lower-quality Speaker instructions.

Gibson, SE3DS and model size Augmentation with Marky instructions in Gibson environments (row 6 v. 3) provides a substantial boost (+11% SR on RxR and +12% on R2R), suggesting that the returns from scaling synthetic instructions to more environments are not exhausted. Using SE3DS to synthesize image observations from novel viewpoints improves +6% SR on RxR and +12% on R2R (row 5 vs. 3), but this benefit is substantially reduced (+0% SR on RxR and +2% on R2R, row 7 vs. 6) if Gibson is included, presumably because new environments also increase viewpoint variety. Most experiments use the mT5-base (Xue et al., 2021) model; switching to mT5-large provides a further performance boost (+2% SR on RxR and +1% on R2R, row 8 vs. 7). Our best pretraining results on both RxR and R2R are achieved using an mT5-large model with all the previously mentioned data, but leaving out the Speaker instructions (row 9). We use this checkpoint in all finetuning experiments. This agent pretrains for 5.14M iterations, which, using a batch size of 128, represents over 650M steps of experience (over 700M including finetuning).

Finetuning In Tables 3 and 4 we compare results for our MARVAL agent after finetuning to previous work on the RxR and R2R datasets. On both datasets, finetuning with behavioral cloning on just human-annotated data (Finetuned-BC) substantially improves the pretrained model. The improvement from using DAGGER over behavioral cloning is small but consistent. On the RxR dataset, MARVAL outperforms all prior work. Evaluating on novel instruction-trajectories in seen environments (Val-Seen), we improve over the state-of-the-art by 8%, reaching 79.1 NDTW. In new, unseen environments (Test), we improve by 2%, achieving 66.8 NDTW. Self-training with Marky synthetic instructions in the Test environments (a form of privileged access, but still without human annotations) improves performance by an additional 2% to 68.6 NDTW.

	Val-Seen				**Val-Unseen**				**Test (Unseen)**			
Agent	**NE**	**SR**	**NDTW**	**SDTW**	**NE**	**SR**	**NDTW**	**SDTW**	**NE**	**SR**	**NDTW**	**SDTW**
LSTM (Ku et al., 2020)	10.7	25.2	42.2	20.7	10.9	22.8	38.9	18.2	12.0	21.0	36.8	16.9
EnvDrop+ (Shen et al., 2021)	-	-	-	-	-	42.6	55.7	-	-	38.3	51.1	32.4
CLEAR-C (Li et al., 2022b)	-	-	-	-	-	-	-	-	-	40.3	53.7	34.9
HAMT (Chen et al., 2021)	-	59.4	65.3	50.9	-	56.5	63.1	48.3	6.2	53.1	59.9	45.2
EnvEdit* (Li et al., 2022a)	-	67.2	71.1	58.5	-	62.8	68.5	54.6	**5.1**	60.4	64.6	51.8
MARVAL (Pretrained)	3.62	72.7	77.0	65.9	5.56	59.4	67.0	52.7	-	-	-	-
MARVAL (Finetuned-BC)	3.25	75.4	79.0	68.7	4.80	63.7	70.6	56.9	-	-	-	-
MARVAL (DAGGER)	**3.01**	**75.9**	**79.1**	**68.8**	**4.49**	**64.8**	**70.8**	**57.5**	5.5	**60.7**	**66.8**	**53.5**
MARVAL (Pre-Explore)†	3.33	73.7	77.6	66.6	4.19	66.5	72.2	59.1	5.2	61.8	68.6	54.8
Human (Ku et al., 2020)	-	-	-	-	-	-	-	-	0.9	93.9	79.5	76.9

*Results from an ensemble of three agents.

Table 3: Results on RxR. Our MARVAL agent trained with imitation learning – behavioral cloning (BC) or DAGGER – outperforms all existing RL agents. Pre-Exploration in the eval environments († a form of privileged access, but still without human annotations) can provide a further boost.

	Val-Seen				**Val-Unseen**				**Test (Unseen)**			
Agent	**TL**	**NE**	**SR**	**SPL**	**TL**	**NE**	**SR**	**SPL**	**TL**	**NE**	**SR**	**SPL**
PREVALENT (Hao et al., 2020)	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
RecBERT (Hong et al., 2021)	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
EnvDrop+ (Shen et al., 2021)	-	-	-	-	-	-	59.2	52.9	-	-	59	53
AirBERT (Guhur et al., 2021)	11.09	2.68	75	70	11.78	4.01	62	56	12.41	4.13	62	57
HAMT (Chen et al., 2021)	11.15	2.51	76	72	11.46	2.29	66	61	12.27	3.93	65	60
REM (Liu et al., 2021)	10.88	2.48	75.4	71.8	12.44	3.89	63.6	57.9	13.11	3.87	65	59
EnvEdit* (Li et al., 2022a)	11.18	**2.32**	**76.9**	**73.9**	11.13	**3.24**	**68.9**	**64.4**	11.90	**3.59**	**68**	**64**
MARVAL (Pretrained)	10.32	3.73	68.2	64.9	9.71	4.84	57.4	54.6	-	-	-	-
MARVAL (Finetuned-BC)	10.43	3.11	72.3	68.9	9.72	4.20	63.0	60.0	-	-	-	-
MARVAL (DAGGER)	10.60	2.99	73.0	69.1	10.15	4.06	64.8	60.7	10.22	4.18	62	58
Human (Anderson et al., 2018b)	-	-	-	-	-	-	-	-	11.90	1.61	86	76

*Results from an ensemble of three agents.

Table 4: Results on R2R. MARVAL achieves strong performance but not state-of-the-art, which we attribute to domain differences between the R2R and RxR (which was used to train Marky).

RxR vs. R2R On the English-only R2R dataset (Table 4), MARVAL achieves strong performance but not state-of-the-art. Surprisingly, the Val-Unseen success rate (SR) of 64.8% is the same for both RxR and R2R, whereas typically RxR performance is lower since the trajectories are longer and more varied. Noting that Marky was trained on RxR, we attribute lower relative performance on R2R to domain differences between R2R and RxR. While the average length of instructions in R2R is 26 words, RxR has an average of 87 words — 3 times more. This is partly because RxR instructions are more verbose, often describing objects in more detail and including state verification. Further, cultural differences arising from the data collection process (annotators from USA or from India) may also contribute to the distribution shift due to subtle differences in the vocabulary and structure of language used to form the instructions. We note however, that while our augmentation approach focuses on scaling up in terms of high quality instructions, EnvEdit (Li et al., 2022a) focuses on generalization through augmentation of visual features. These two approaches are likely to be complementary and can provide further improvements on both R2R and RxR.

6 Conclusion

To our knowledge, this is the first time an imitation learning agent has achieved state-of-the-art or even competitive results on the RxR or R2R benchmarks. This result paves a new path towards improving instruction-following agents, emphasizing large-scale imitation learning with generic architectures, along with a focus on developing synthetic instruction generation capabilities – which are shown to flow through directly to improved instruction-following performance.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On Evaluation of Embodied Navigation Agents. *ArXiv preprint*, 2018a.

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

Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *CoRL*, 2021.

Jacob Andreas and Dan Klein. Alignment-based compositional semantics for instruction following. In *EMNLP*, 2015.

Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *TACL*, 2013.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *EMNLP*, 2020.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *3DV*, 2017.

David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *ICCV*, 2019.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. *ArXiv preprint*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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

Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. *ICCV*, 2021.

Meera Hahn, Jacob Krantz, Dhruv Batra, Devi Parikh, James Rehg, Stefan Lee, and Peter Anderson. Where are you? localization from embodied dialog. In *EMNLP*, 2020.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

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

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. VLNBERT: A recurrent vision-and-language bert for navigation. *CVPR*, 2021.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. Effective and general evaluation for instruction conditioned navigation using dynamic time warping. *NeurIPS Visually Grounded Interaction and Language Workshop*, 2019.

Aashi Jain, Mandy Guo, Krishna Srinivasan, Ting Chen, Sneha Kudugunta, Chao Jia, Yinfei Yang, and Jason Baldridge. MURAL: Multimodal, multitask representations across languages. In *EMNLP Findings*, 2021.

Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *ArXiv preprint*, 2018.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. *ICCV*, 2021.

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

Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Simple and effective synthesis of indoor 3d scenes. *ArXiv preprint*, 2022.

Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 1956.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, 2018.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

Jialu Li, Hao Tan, and Mohit Bansal. EnvEdit: Environment editing for vision-and-language navigation. In *CVPR*, 2022a.

Jialu Li, Hao Tan, and Mohit Bansal. CLEAR: Improving vision-language navigation with cross-lingual, environment-agnostic representations. In *Findings of the Association for Computational Linguistics: NAACL*, 2022b.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022c.

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

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

Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, and Yi-Dong Shen. Vision-language navigation with random environmental mixup. *ICCV*, 2021.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*, 2019.

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *ECCV*, 2020.

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

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *AAAI*, 2016.

Dipendra Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. In *EMNLP*, 2017.

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

Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene- and object-aware transformer for vision-and-language navigation. *NeurIPS*, 2021.

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.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *AISTATS*, 2010.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

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

Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied AI research. In *ICCV*, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? *ArXiv*, 2021.

Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL-HLT*, 2019.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, Proceedings of Machine Learning Research, 2019.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, 2019.

Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks. In *CVPR*, 2022.

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

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. *ArXiv*, 2021.

Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical report, Massachusetts Institute of Technology, 1971.

Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2021.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *ArXiv preprint*, 2022.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. *CVPR*, 2021.

Yubo Zhang, Hao Tan, and Mohit Bansal. Diagnosing the environment bias in vision-and-language navigation. In *IJCAI*, 2020.

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

Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Eric Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. Diagnosing Vision-and-Language Navigation: What Really Matters. In *NAACL-HLT*, 2022.

A Appendix

A.1 Limitations and Future Work

As we discuss in the main paper, our approach achieves strong but not state-of-the-art results on the R2R dataset, which we attribute to domain differences between R2R and RxR (noting that the Marky instruction generator we use for data augmentation was trained on RxR). One way to address this limitation would be by re-training Marky on R2R data, although this would face some hurdles since R2R lacks the annotator pose traces that were used by Marky when training on RxR.

To better understand the failure modes of our approach, in Figure 3 we plot a distribution indicating the step in the trajectory where an agent makes its first error. We analyze the RxR Val-Unseen split and compare MARVAL to the previous state-of-the-art approaches, EnvEdit and HAMT, as well as human instruction-following demonstrations from the RxR dataset. MARVAL makes fewer errors than the prior approaches, especially at the start of the trajectory, but also *fewer errors than human followers*. Since human followers still significantly outperform MARVAL overall – in terms of navigation error (0.79m vs. 4.49m), success rate (94.5 vs. 64.8) and path-fidelity metrics such as NDTW (81.8 vs. 70.8) – this suggests the main focus for future agent improvement should be on recovering from errors, which human wayfinders clearly do extremely well in order to still reach the goal in 94.5% of episodes.

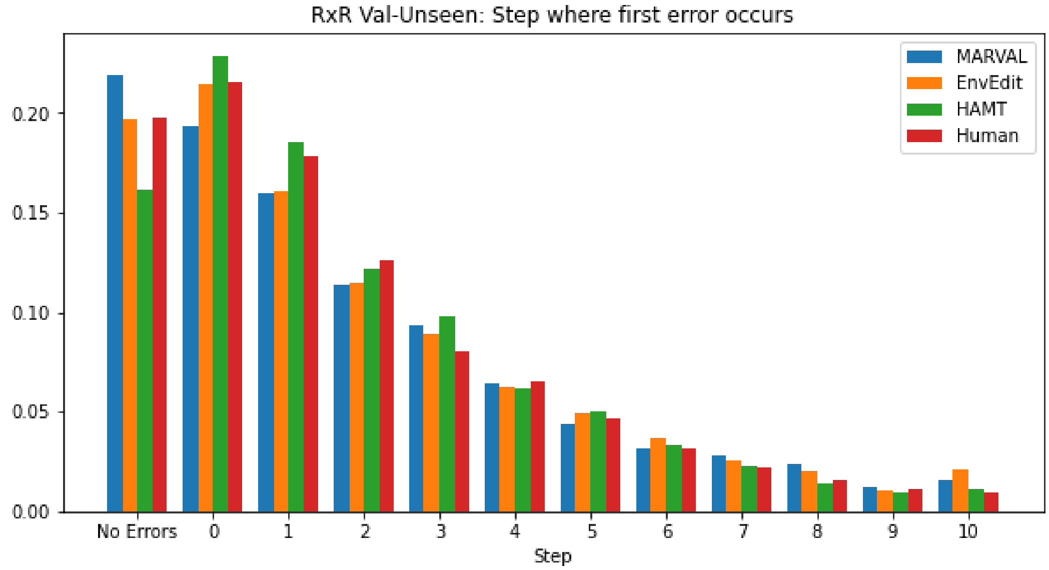


Figure 3: Error analysis indicating at which trajectory step each agent makes its first mistake. Surprisingly, MARVAL makes less errors (produces more perfect trajectories) than the prior work *and human followers*. Since human followers still significantly outperform MARVAL overall, this suggests the main focus for future agent improvement should be on recovering from errors.

A.2 Implementation Details

Pretraining In all experiments we train with a batch size of 128 using the AdaFactor optimizer. During pretraining, we use dropout of 0.1 and a learning rate that exponentially-decays from 0.1. We monitor one-step action prediction accuracy on ground-truth trajectories using held-out instructions from unseen environments (RxR Val-Unseen and R2R Val-Unseen). We pretrain until convergence and then select the best snapshot based on one-step action prediction accuracy on RxR Val-Unseen.

Finetuning During finetuning, we use a constant learning rate of 0.001 and dropout of 0.2. Since the human-annotated datasets used for finetuning (RxR and R2R) are relatively small, during finetuning we update only the WordPiece embeddings in the agent and keep all other transformer weights frozen. This makes finetuning more stable and less prone to overfitting (especially on the smaller

R2R dataset). We finetune for a maximum of 150K iterations while monitoring standard VLN path-fidelity metrics such as success rate (SR). We select the best snapshot based on SR on Val-Unseen.

A.3 Pretraining Results

In Table 5 we report pretraining results on the *Val-Seen* splits, complementing the *Val-Unseen* results included in the main paper. We observe the same trends in the seen environments as in the new, unseen environments (Val-Unseen), although the relative improvement from using a larger model (row 8 vs. 7) is larger.

		Pretraining Data							**RxR Val-Seen**				**R2R Val-Seen**		
	Size	**R2R**	**RxR**	**S-MP**	**M-MP**	**M-Gib**	**SE3DS**	**Steps**	**NE**	**SR**	**NDTW**	**SDTW**	**NE**	**SR**	**SPL**
1	Base	✓	✓					630K	12.14	22.3	39.3	18.8	7.45	37.9	36.3
2	Base	✓	✓	✓				1.68M	11.52	26.7	42.4	22.9	5.19	52.6	50.5
3	Base	✓	✓	✓	✓			2.94M	7.23	48.5	59.3	42.4	6.19	48.7	45.7
4	Base	✓	✓		✓			2.00M	4.42	64.8	73.2	58.5	3.86	67.2	64.2
5	Base	✓	✓	✓	✓		✓	1.94M	5.15	62.0	69.9	55.6	4.61	59.2	56.2
6	Base	✓	✓	✓	✓	✓		3.00M	5.09	61.4	68.7	51.2	4.77	57.5	53.0
7	Base	✓	✓	✓	✓	✓	✓	2.80M	5.22	60.0	68.3	53.1	4.70	59.6	55.1
8	Large	✓	✓	✓	✓	✓	✓	4.80M	3.85	71.0	75.9	64.3	4.05	66.1	62.6
9	Large	✓	✓		✓	✓	✓	5.14M	**3.62**	**72.7**	**77.0**	**65.9**	**3.73**	**68.2**	**64.9**

Table 5: Comparison of pretraining approaches reporting Val-Seen results on RxR and R2R.

A.4 Performance on different languages

	Val-Seen				**Val-Unseen**			
Agent	**NE ↓**	**SR↑**	**NDTW↑**	**SDTW↑**	**NE↓**	**SR↑**	**NDTW↑**	**SDTW↑**
				en-IN				
EnvEdit* (Li et al., 2022a)	3.82	68.01	71.45	59.09	4.42	62.03	67.89	54.15
MARVAL	3.10	75.77	78.79	68.86	4.49	64.83	70.67	57.64
				en-US				
EnvEdit* (Li et al., 2022a)	4.22	66.19	69.52	56.95	4.33	61.70	67.56	52.94
MARVAL	3.53	72.22	76.19	64.62	4.46	64.47	70.28	56.46
				te-IN				
EnvEdit* (Li et al., 2022a)	3.72	65.59	70.70	57.37	4.49	61.85	67.84	53.75
MARVAL	2.82	76.38	79.34	69.43	4.56	63.85	69.92	56.30
				hi-IN				
EnvEdit* (Li et al., 2022a)	4.01	68.56	71.90	59.92	4.20	64.54	69.74	56.41
MARVAL	2.95	76.61	80.05	69.52	4.30	65.85	72.02	58.88

*Results from an ensemble of three agents.

Table 6: Breakdown of the results on RxR for each language for our best model and the best performing baseline model from Li et al. (2022a). Predicted paths for EnvEdit were provided by the authors.